

AFIT/GE/ENG/91D-48

DTIC
ELECTE
DEC 27 1991
S C D

AD-A243 833



Principal Base Parameter Analysis: Implementation and Analysis in an Adaptive
Model-Based Robotic Controller

THESIS

Gregory L. Showman
Captain, USAF

AFIT/GE/ENG/91D-48

Approved for public release; distribution unlimited

91-19025



91 12 24 047

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
This report contains information which may be subject to copyright. No part of this report may be reproduced without permission in writing from the copyright owner. Send comments regarding this burden estimate or any other aspect of this report, including suggestions for reducing the burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1991	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE Principal Base Parameter Analysis: Implementation and Analysis in an Adaptive Model-Based Robotic Controller			5. FUNDING NUMBERS	
6. AUTHOR(S) Gregory L. Showman, Captain, USAF			7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology, WPAFB OH 45433-6583	
8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GE/ENG/91D-48			9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES) Armstrong Laboratory/BBA, WPAFB OH 45433-6573	
10. SPONSORING MONITORING AGENCY REPORT NUMBER			11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Principal Base Parameter Analysis (PBPA) is a general and systematic procedure for determining the dynamic parameters that directly contribute to the joint torques of a manipulator, ranked in order of sensitivity. The feasibility of employing PBPA as an aid in the design and tuning of adaptive model-based controllers for industrial manipulators is rigorously investigated. This is accomplished by employing PBPA to determine the minimal size of the adaptive parameter vector and more importantly, to develop a less heuristic procedure for controller tuning. A simple, step-by-step procedure is developed wherein the manipulator torque equations are used in conjunction with PBPA to develop a functional adaptive model-based control (AMBC) algorithm, then tune the algorithm for optimal performance. Experimental analysis contrasts this adaptive model-based controller, designed and tuned using PBPA, to the completely heuristic procedure employed in previous Air Force Institute of Technology research. The incorporation of PBPA into the AMBC design methodology reduces the time and expertise necessary to tune the controller for satisfactory tracking performance.				
14. SUBJECT TERMS Adaptive Model-Base Control, Parameter Tuning, Parameter Reduction, Parameter Analysis, Robotic Control, Adaptive Control			15. NUMBER OF PAGES 132	
16. PRICE CODE			17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	
18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED			19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	
20. LIMITATION OF ABSTRACT UL			21. LIMITATION OF ABSTRACT UL	

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet optical scanning requirements.

Block 1. Agency Use Only (Leave blank)

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	PR - Project
G - Grant	TA - Task
PE - Program Element	WU - Work Unit Accession No.

Block 6. Author(s) Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as. Prepared in cooperation with..., Trans. of..., To be published in... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."

DOE - See authorities.

NASA - See Handbook NHB 2200.2.

NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.

DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

NASA - Leave blank.

NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

AFIT/GE/ENG/91D-48

Principal Base Parameter Analysis: Implementation and Analysis in
an Adaptive Model-Based Robotic Controller

THESIS

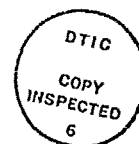
Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Electrical Engineering

Gregory L. Showman, BSEE
Captain, USAF

12 December 1991



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Approved for public release; distribution unlimited

Acknowledgements

An effort such as this involves many more people than just the author. First, I would like to thank Dr. Michael B. Leahy Jr., who identified the need for this area of research and kept me from straying too far afield. He has always presented a superlative example of both an officer and an engineer. Also, a special thanks to my committee members, Dr. Curtis Spenny and Dr. Matthew Kabrisky for their insights into my research.

Just as important are those who were always there for everyday problems and questions. Thanks to Dan Zambon who kept the computers up despite the efforts of those with enough knowledge to be dangerous. Also, thanks to Paul Whalen who saved me from many hours of chasing my tail.

Finally, and most of all, I would like to thank the best family in the world - my AFIT-widow Beatrice and three kids Ryan, Abigail and Adrienne. This effort would never have been possible without all their love, support and understanding.

Table of Contents

	Page
Table of Contents	iii
List of Figures	iv
List of Tables	v
Abstract	vi
 I. Problem Description	 1-1
1.1 Background	1-1
1.2 Problem Statement and Objective	1-2
1.3 Application of Principal Base Parameter Analysis	1-3
1.4 Method of Attack	1-4
1.4.1 Incorporation of PBPA into an AMBC algorithm	1-5
1.4.2 Experimental Verification and Validation of Technique	1-5
1.4.3 Cross Check of Technique on a Second Platform	1-6
1.5 Materials and Equipment	1-6
1.6 Contributions	1-7
1.7 Summary	1-8
 II. Literature Review	 2-1
2.1 Introduction	2-1
2.2 Adaptive Model-Based Control	2-1
2.2.1 AFIT Research	2-3
2.3 Parameter Reduction	2-7

	Page
2.3.1 Base Parameters of Manipulator Dynamic Models	2-9
2.3.2 Direct Calculation of the Minimum Set of Inertial Parameters	2-10
2.3.3 Principal Based Parameter Analysis	2-10
2.3.4 Development of PBPA Equations	2-11
2.4 Conclusion	2-14
III. Adaptive Model-Based Control Development using Principal Base Parameter Analysis	3-1
3.1 Overview	3-1
3.2 AMBC Design with PBPA	3-1
3.2.1 Ideal 2 DOF PUMA	3-2
3.2.2 The Three DOF Case	3-6
3.3 The Utah/MIT Dexterous Hand Implementation	3-8
3.4 Summary	3-8
IV. PBPA Test and Analysis	4-1
4.1 Overview	4-1
4.2 Test Environment	4-1
4.2.1 Description of Trajectories	4-2
4.2.2 Initialization of \hat{a}	4-4
4.2.3 Γ^{-1} Values	4-6
4.3 Analysis of Data	4-7
4.3.1 Evaluation Over Standard Trajectories	4-7
4.3.2 Adaption of \hat{a}	4-14
4.3.3 Robustness to Payload Variation	4-15
4.3.4 Decreasing Number of Estimated Parameters	4-15

	Page
4.3.5 Importance of Accurate Knowledge of Physical Values	4-21
4.3.6 Adaptation versus Pattern Learning	4-24
4.3.7 Soft PD Gains	4-24
4.3.8 Two Way Tracking Evaluation	4-29
4.3.9 Very Slow Tracking	4-29
4.3.10 Effect of Varying the Γ^{-1} Scaling Factor . . .	4-29
4.3.11 Numerical Comparison	4-32
4.4 The Utah/MIT Dexterous Hand	4-35
4.5 Summary	4-36
V. Conclusions and Recommendations	5-1
5.1 Conclusions	5-1
5.2 Recommendations for Future Study	5-1
5.3 Disparity Between AMBC/H and AMBC/PBPA	5-2
5.4 AMBC/PBPA Fine-Tuning	5-2
5.5 Cross Check of Procedure on a Second Platform	5-3
5.6 Summary	5-3
Appendix A. Comprehensive Set of Figures	A-1
Appendix B. PUMA-560, Three Degree of Freedom Values	B-1
B.1 The Non-linear Function Vector, n_0	B-1
B.2 Constant Matrices of Kinematic Parameters, k_i	B-2
B.3 The Non-reduced Parameter Set, p	B-4
B.4 The AMBC/PBPA Regressor Matrix	B-4
B.5 The Principal Base Parameter Set, P^*	B-9
B.6 Physical Values Used	B-10
Bibliography	BIB-1

List of Figures

Figure	Page
2.1. PD Only vs PD with AMBC - Trajectory 1	2-6
2.2. Effect of setting AMBC/H Γ^{-1} to a Common Value: Fifth Run, Trajectory 1	2-8
4.1. AMBC/H vs. AMBC/PBPA - Trajectory 1	4-8
4.2. AMBC/H vs. AMBC/PBPA - Trajectory 2	4-9
4.3. AMBC/H vs. AMBC/PBPA - Trajectory 3	4-10
4.4. AMBC/H vs. AMBC/PBPA - Trajectory 5	4-11
4.5. AMBC/H vs. AMBC/PBPA - Trajectory 6	4-12
4.6. Illustration that AMBC/H and AMBC/PBPA do not converge - Trajectory 3	4-13
4.7. AMBC/H vs. AMBC/PBPA - Trajectory 1 with 2KG Payload .	4-16
4.8. AMBC/H vs. AMBC/PBPA - Trajectory 3 with 2KG Payload .	4-17
4.9. Effect of Parameter Reduction: Fifth Run, Trajectory 1	4-18
4.10. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2	4-19
4.11. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3	4-20
4.12. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2	4-22
4.13. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3	4-23
4.14. Effect of Varying Physical Parameter Values: Trajectory 1, Run 5	4-25
4.15. Effect of Varying Physical Parameter Values: Trajectory 3, Run 5	4-26
4.16. Effect of Discontinuing Adaptation - Trajectory 1	4-27
4.17. Effect of Soft PD Gains, Trajectory 1, No Payload	4-28

Figure	Page
4.18. Two Way Trajectory Tracking Evaluation, No Payload, Trajectory 1	4-30
4.19. Very Slow Tracking Evaluation, No Payload, Trajectory 3	4-31
4.20. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 1	4-33
4.21. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 5	4-34
A.1. PD Only vs PD with AMBC/H - Trajectory 1	A-2
A.2. Effect of setting AMBC/H Γ^{-1} to a Common Value: Fifth Run, Trajectory 1	A-2
A.3. AMBC/H vs. AMBC/PBPA- Trajectory 1	A-3
A.4. AMBC/H vs. AMBC/PBPA - Trajectory 2	A-3
A.5. AMBC/H vs. AMBC/PBPA - Trajectory 3	A-4
A.6. AMBC/H vs. AMBC/PBPA - Trajectory 4	A-4
A.7. AMBC/H vs. AMBC/PBPA - Trajectory 5	A-5
A.8. AMBC/H vs. AMBC/PBPA - Trajectory 6	A-5
A.9. Demonstration of Whether or Not AMBC/H and AMBC/PBPA Ever Become Equal: Run 1, Traj 3	A-6
A.10. Demonstration of Whether or Not AMBC/H and AMBC/PBPA Ever Become Equal: Run 5, Traj 3	A-6
A.11. Demonstration of Whether or Not AMBC/H and AMBC/PBPA Ever Become Equal, Traj 3	A-7
A.12. AMBC/H vs. AMBC/PBPA - Trajectory 1 with 2KG Payload .	A-7
A.13. AMBC/H vs. AMBC/PBPA - Trajectory 3 with 2KG Payload .	A-8
A.14. Effect of Parameter Reduction: First Run, Trajectory 1	A-8
A.15. Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 1	A-9
A.16. Effect of Parameter Reduction: Fifth Run, Trajectory 1	A-9
A.17. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 1	A-10

Figure	Page
A.18.Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 2	A-10
A.19.Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 2	A-11
A.20.Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2	A-11
A.21.Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2	A-12
A.22.Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 3	A-12
A.23.Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 3	A-13
A.24.Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3	A-13
A.25.Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3	A-14
A.26.Effect of Varying Physical Parameter Values: Trajectory 1, Run 1	A-14
A.27.Effect of Varying Physical Parameter Values: Trajectory 1, Run 5	A-15
A.28.Effect of Varying Physical Parameter Values: Trajectory 2, Run 1	A-15
A.29.Effect of Varying Physical Parameter Values: Trajectory 2, Run 5	A-16
A.30.Effect of Varying Physical Parameter Values: Trajectory 3, Run 1	A-16
A.31.Effect of Varying Physical Parameter Values: Trajectory 3, Run 5	A-17
A.32.Effect of Discontinuing Adaptation - Trajectory 1	A-17
A.33.Effect of Discontinuing Adaptation - Trajectory 3	A-18
A.34.Effect of Soft PD Gains, Trajectory 1, No Payload	A-18
A.35.Effect of Soft PD Gains, Trajectory 1, 2KG Payload	A-19
A.36.Effect of Soft PD Gains, Trajectory 3, No Payload	A-19
A.37.Effect of Soft PD Gains, Trajectory 3, 2KG Payload	A-20
A.38.Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 1, No Payload	A-20

Figure	Page
A.39.Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 1, 2KG Payload	A-21
A.40.Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 3, No Payload	A-21
A.41.Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 3, 2KG Payload	A-22
A.42.Two Way Trajectory Tracking Evaluation, No Payload, Trajectory 1	A-22
A.43.Very Slow Tracking Evaluation, No Payload, Trajectory 1	A-23
A.44.Very Slow Tracking Evaluation, No Payload, Trajectory 2	A-23
A.45.Very Slow Tracking Evaluation, No Payload, Trajectory 3	A-24
A.46.Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 1	A-24
A.47.Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 5	A-25
A.48.Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 3, Run 1	A-25
A.49.Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 3, Run 5	A-26

List of Tables

Table	Page
4.1. Stiff PD Feedback Gains	4-2
4.2. Test Trajectories	4-3
4.3. Comparison of \hat{a} Values; 16 Estimated Parameters	4-14
4.4. Comparison of \hat{a} Values; 7 Estimated Parameters	4-21
4.5. Soft PD Feedback Gains	4-29
4.6. Comparison of Tracking Errors using Equation 4.5	4-35

Abstract

Principal Base Parameter Analysis (PBPA) is a general and systematic procedure for determining the dynamic parameters that directly contribute to the joint torques of a manipulator, ranked in order of sensitivity. The feasibility of employing PBPA as an aid in the design and tuning of adaptive model-based controllers for industrial manipulators is rigorously investigated. This is accomplished by employing PBPA to determine the minimal size of the adaptive parameter vector and more importantly, to develop a less heuristic procedure for controller tuning. A simple, step-by-step procedure is developed wherein the manipulator torque equations are used in conjunction with PBPA to develop a functional adaptive model-based control (AMBC) algorithm, then tune the algorithm for optimal performance. Experimental analysis contrasts this adaptive model-based controller, designed and tuned using PBPA, to the completely heuristic procedure employed in previous Air Force Institute of Technology research. The incorporation of PBPA into the AMBC design methodology reduces the time and expertise necessary to tune the controller for satisfactory tracking performance.

Principal Base Parameter Analysis: Implementation and Analysis in an Adaptive Model-Based Robotic Controller

I. Problem Description

1.1 Background

The Air Force must be able to sustain operations in environments ranging from the sub-zero temperatures of Thule, Greenland to the deserts of the Persian Gulf. In addition, during war time, we may need to operate in a chemical, biological or nuclear environment. In these situations, it may be undesirable or impossible to use human operators. Robots may be the answer to this problem; however, if robots are to be used in demanding Air Force applications, further research is necessary in the area of advanced control algorithms.

One existing method of controlling robots, generically called classical control, is to feed back position and velocity information into the control circuitry which causes a modification of the control torque. Many classical control algorithms can supply fairly accurate positioning, but only over specifically defined trajectories. A more serious drawback of classical control algorithms is that they tend to either suffer degraded tracking accuracy or become entirely unstable in the presence of external disturbances or variable payloads. In an effort to make robots more general purpose, as well as consistently accurate over varied conditions, different control algorithms are being researched. One of the more promising classes of algorithms under investigation is Adaptive Model-Based Control (AMBC). Unlike classical methods of control, which rely on well defined manipulator dynamics, AMBC uses an estimated system model and modifies feedforward motor torques based on position and velocity errors. Furthermore, the estimated system model is refined with each successive

pass over a given trajectory. In other words, this type of algorithm has the ability to adapt to changing environments.

Previous studies at the Air Force Institute of Technology have demonstrated that the tracking accuracy of a robot can show significant improvement when Adaptive Model-Based Control is used [16, 22]. Unfortunately, in order to achieve optimal performance, these studies have had to employ heuristic, manpower intensive methods to fine tune the algorithm. The prime objective of this study has been to move the tuning process away from this iterative, experimental nature.

The remainder of this chapter will be devoted to exactly what the problem is, as well as how it was addressed. First, the problem will be explored more in depth, both in terms of what the problem attributes are and why this subject rates further study. Next, a general outline of the approach and methodology of this study will be presented. This roadmap of the research will cover not only what was done and what physical resources were used, but also what the final goal was.

1.2 Problem Statement and Objective

As already stated, previous studies at the Air Force Institute of Technology (AFIT) have demonstrated that the tracking accuracy of a robot can show significant improvement over simple feedback controllers when Adaptive Model-Based Control (AMBC) is used [16, 22]. For a typical AMBC algorithm, the total joint torque applied is the feedback (i.e. Proportional-Derivative or PD) torque added to a feedforward torque. This feedforward torque is typically determined real-time by the adaptive algorithm. Specifically, the feedforward torque is determined on a joint-by-joint basis and overall, is a product of a regressor matrix (Y) and a parameter vector (\hat{a}). The regressor matrix is comprised of all the non-linear terms of the manipulator torque equations while the parameter vector is made up of the known dynamic terms and an estimate of the unknown dynamic terms. The estimated portion of \hat{a} is the product of the regressor transposed, the position and velocity errors and a

diagonal adaptive gain matrix, Γ^{-1} all integrated over time. These terms, as well as their usage, will be discussed more in depth in the following chapters. The adaptive gain matrix, Γ^{-1} , is what was actually tuned with the aforementioned heuristic, manpower intensive tuning process. One of the previous AFIT studies, done by Leahy and Whalen, used a Γ^{-1} of 16 diagonal elements, each ranging in value from 0 to 150 [16]. Assuming complete independence of the elements, total possible combinations of values approaches infinity. For the Leahy and Whalen AMBC development, the tuning was performed over the course of months, by individuals very well versed in the dynamics of that particular manipulator. Leahy and Whalen have stated that the tuning process is more an art form than a science and that changing the Γ^{-1} values could result in either improved performance or disaster [16].

If it is desired to develop an AMBC algorithm for a general manipulator, the question remains - is there a procedure to develop and tune said algorithm without employing heuristics or extensive a priori knowledge of manipulator dynamics? While it may be acceptable to spend months tuning a specific manipulator for a specific task in a laboratory environment, such a time consuming process negates many of the AMBC benefits in the context of day-to-day operation. Therefore, the objective of this study was to develop a straightforward AMBC design procedure that eliminates or reduces the amount of heuristics used when tuning the algorithm for optimal performance.

1.3 Application of Principal Base Parameter Analysis

One way in which the tuning process could be simplified would be to establish a relationship of each element of the parameter vector to the other elements. Assuming that such a relationship could be found, the parameter vector could then be tuned as a whole, using a single scaling factor, as opposed to the exhaustive combinational analysis described in the previous section. A method of parameter analysis, recently proposed by Ghodoussi and Nakamura, might be able to be used in this application.

Their procedure, Principal Base Parameter Analysis reduces the parameter set of a manipulator to an absolute minimum and ranks the resultant parameter vector elements in order of sensitivity [5]. As a byproduct, PBPA yields an element-to-element relationship in the resultant, reduced parameter vector. Ghodoussi and Nakamura suggest that this element to element relationship may be of use in a control application.

PBPA starts from the fact that each joint of a robot has ten associated primitive dynamic parameters, which describe how it is moved and positioned [20]. These primitive dynamic parameters are the link mass m , the independent elements of the inertia tensor I_{xx} , I_{yy} , I_{zz} , I_{xy} , I_{xz} , I_{yz} and the three position elements of the mass centroid, r_x , r_y , r_z [5]. When moving the end effector into some arbitrary position, some of these parameters are more important than others and some parameters will be redundant. The base parameter set is defined as the minimal set of parameters necessary to fully describe the dynamics of a manipulator. Several methods have been proposed to quantify how many parameters make up the base parameter set [8, 4]; however, the method presented by Ghodoussi and Nakamura not only identifies the minimal parameters specifically, but also ranks them in order of sensitivity. This reduced, ranked parameter set is called the principal base parameter set.

Principal Base Parameter Analysis is a method to reduce the parameter set to a minimum and coincidentally, establishes an element to element relationship in the reduced parameter vector [5]. This study has taken that assumption and used it to develop an AMBC algorithm that can be tuned with a single scaling factor versus the previous heuristic tuning method.

1.4 Method of Attack

The first thrust of this study was to perform PBPA on the PUMA-560 configured as a two degree of freedom platform (actuating only joints two and three). While this portion of the study has little real-world application, it serves to illus-

trate the concepts. The analysis was then expanded to a three degree of freedom case (joints one, two and three actuated). The results obtained via PBPA for the three are then incorporated into an adaptive model-based control algorithm. Finally, to verify that this technique is not platform specific, the same procedure was applied to a totally disparate manipulator, the Utah/MIT Dexterous Hand (UMDH).

1.4.1 Incorporation of PBPA into an AMBC algorithm

As discussed in the preceeding sections, PBPA yields a reduced parameter set, ranked in order of sensitivity. An overview of how this reduced parameter set is incorporated into an AMBC algorithm is as follows; specifics and description of terms will follow in later chapters. First, the reduced parameter vector was used in place of the the linear parameter vector, \hat{a} , in an AMBC algorithm. Next, using the original torque equations for the three degree of freedom PUMA and this parameter vector, a new regressor matrix was determined (such that the product of the regressor and the parameter vector equaled the original torque equations). Finally, since the parameter set consists only of physical values (i.e. link lengths, masses, gravity), the best estimate of physical values were substituted into the reduced parameter vector. This substitution yielded a vector of strictly numerical values. This set of numerical values was then used as the base Γ^{-1} values in the AMBC algorithm. From this point on, all tuning of the algorithm was accomplished by multiplying the Γ^{-1} matrix by a single scaling factor.

1.4.2 Experimental Verification and Validation of Technique

Upon completion of the AMBC design, an exhaustive experimental analysis was performed to provide validation and verification of the procedure. The trajectory set, as detailed by Leahy and Whalen [16], was performed to remain consistant with previous AFIT research. Additionally, this previous AFIT study was used as the benchmark with which to determine success or failure of the AMBC/PBPA design and tuning technique. As was done by Leahy and Whalen, this study also

investigated such items as the effect of payload and parameter reduction on tracking accuracy.

1.4.3 Cross Check of Technique on a Second Robot

In order to prove that this technique of algorithm tuning is not just applicable to the PUMA, PBPA and the associated AMBC tuning was also planned for the Utah/MIT Dexterous Hand. It was hoped that this cross check would prove that the technique developed here could be applied across a range of robots. The study done at AFIT by Rainey had already developed a control algorithm development environment for the UMDH; however, since the time of that study, the host computer and operating system had changed [22]. This change of host precipitated changes in both the code and the communications interface to the robot. Consequently, significant effort was required before PBPA and the associated algorithm tuning could be accomplished on the UMDH. Due to delays in hardware development, the UMDH portion of this effort was not brought to fruition. PBPA was performed on the UMDH and the results were incorporated into an AMBC algorithm; however, no experimental analysis was performed. Therefore, no conclusions can be drawn as to how well this design procedure works on the UMDH.

1.5 Materials and Equipment

This study has been performed using equipment and software resident in the Air Force Institute of Technology Signal Processing Lab. Specifically:

- PUMA-560 Vertically Articulated Robot
- AFIT Robotic Control Algorithm Development and Evaluation (ARCADE) environment
- Sun Sparc2 Workstation
- Mathematica Software [28]

- Utah/MIT Dexterous Hand
- SARCOS Hand Control Electronics [10]
- Ironics IV-3272 System Controller [9]
- Ironics IV-3201 VME-bus Multiprocessing Engine
- Data Translation DT1401 Series A/D and D/A Converters
- VME Chassis
- Sun Sparc2 Workstation
- CHIMERA II Real-time Programming Environment [2]

The PUMA version of ARCADE is hosted on a VAXstation III and has both serial and parallel connections to the original PUMA computer bus. The PUMA's LSI-11/73 serves as a preprocessor. Communications restrictions, minimal processing time and nominal clock rate resulted in a servo rate of 4.5 msec (222 Hz) for the experimental evaluations [16]. The UMDH version of ARCADE was developed and hosted on an IBM PC/AT-386 and interfaced to the UMDH via an IV3201 real-time processing engine by IRONICS [22]. The AMBC algorithm required a minimum of 3.0 msec (333 Hz) servo rate due to access and set-up times for the A/D and D/A convertors. As a first step to performing and testing the PBPA design procedure on the UMDH, the previously developed software was re-hosted to a Sparc2 workstation with the CHIMERA II programming environment. This re-hosting proceeded only to the point where the AMBC software was compiled. The continuation of this effort will be the subject of future study.

1.6 Contributions

AMBC type controllers have demonstrated greatly improved tracking accuracy over simple feedback controllers alone. However, the manpower intensive tuning that this class of algorithm requires to achieve optimal performance has made AMBC

unattractive for use in other than a laboratory environment. This study develops and verifies a simple step-by-step procedure by which an AMBC algorithm can be implemented and tuned for optimal tracking accuracy. Using this procedure, an AMBC algorithm can be tuned for different applications in a matter of hours, versus the current time of months. Furthermore, this tuning can be done by a person with little or no knowledge of the manipulator dynamics. The procedure developed herein is the first step in moving AMBC out of the laboratory and into the field.

1.7 Summary

This chapter has described the problems associated with AMBC type algorithms and how this study proposes to solve them. Chapter 2 will explore current literature for research trends in this area. Chapter 3 will describe the AMBC design procedure and illustrate the same with an example. Experimental results are contained in chapter 4 and the final chapter contains conclusions and recommendations for future study.

II. Literature Review

2.1 Introduction

Recent research throughout the robotics community has extensively investigated the topic of Adaptive Model-Based Control (AMBC). Two studies recently completed at the Air Force Institute of Technology (AFIT) have implemented AMBC algorithms on both the PUMA-560 vertically articulated robot [16] and the Utah/MIT Dexterous Hand (UMDH) [22]. These studies both demonstrated that the tracking accuracy of a robot can show significant improvement over simple feedback controllers when an Adaptive Model-Based Control algorithm is used. A problem encountered in both of the forementioned AFIT studies was the method in which the AMBC algorithm was fine tuned for optimal performance. In both cases, heuristic, manpower intensive methods were used to tune the algorithm as well as to reduce the parameter set. The purpose of this literature review is to quantify the current state of research in the areas of AMBC development, tuning of the AMBC algorithms for optimal performance as well as parameter set reduction techniques.

2.2 Adaptive Model-Based Control

For a typical AMBC algorithm, the total joint torque applied is the feedback (i.e. Proportional-Derivative or PD) torque added to a feedforward torque.

$$\tau_{total} = \tau_{fb} + \tau_{ff} \quad (2.1)$$

This feedforward torque is typically determined real-time by the adaptive algorithm. Specifically, the feedforward torque is a product of a regressor matrix (Y) and a parameter vector (\hat{a}). The feedforward torque equation may be of the form:

$$\tau_{ff} = Y(q_d, \dot{q}_d, \ddot{q}_d)\hat{a} \quad (2.2)$$

where

$$Y(q_d, \dot{q}_d, \ddot{q}_d) = \text{regressor matrix} \quad \text{where} \quad Y \in R^{N_{joints} \times N_{parameters}}. \quad (2.3)$$

$$\hat{a} = \text{linear parameter vector} \quad \text{where} \quad \hat{a} \in R^{N_{parameters} \times 1}. \quad (2.4)$$

$$q_d = \text{desired position} \quad (2.5)$$

$$\dot{q}_d = \text{desired velocity} \quad (2.6)$$

$$\ddot{q}_d = \text{desired acceleration} \quad (2.7)$$

The regressor matrix is comprised of all the non-linear terms of the manipulator torque equations while the the parameter vector is made up of the known dynamic terms and an estimate of the unknown dynamic terms (e.g. gravity, inertias, masses, link lengths). For a general 2 degree of freedom case the regressor would be [21]:

$$Y(q_d, \dot{q}_d, \ddot{q}_d) = \quad (2.8)$$

$$\begin{bmatrix} \ddot{q}_{d1} & \ddot{q}_{d2} & \ddot{q}_{d1} + \ddot{q}_{d2} & 2\cos q_{d2} \ddot{q}_{d1} + \cos q_{d2} \ddot{q}_{d2} - 2\sin q_{d2} \dot{q}_{d1} \dot{q}_{d2} - \sin q_{d2} \dot{q}_{d2}^2 \\ 0 & 0 & \ddot{q}_{d1} + \ddot{q}_{d2} & \cos q_{d2} \ddot{q}_{d1} + \sin q_{d2} \dot{q}_{d2}^2 \\ \ddot{q}_{d1} & \ddot{q}_{d1} + \ddot{q}_{d2} & \cos q_{d1} & \cos q_{d1} & \cos(q_{d1} + q_{d2}) & q_{d1} & 0 & \text{sgn} q_{d1} & 0 \\ 0 & \ddot{q}_{d1} + \ddot{q}_{d2} & 0 & 0 & \cos(q_{d1} + q_{d2}) & 0 & q_{d2} & 0 & \text{sgn} q_{d2} \end{bmatrix}$$

where $\text{sign}(x)$ is defined as:

$$\text{sgn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases} \quad (2.9)$$

Equation 2-2 can also be written as

$$\tau_{ff} = Y_1(q_d, \dot{q}_d, \ddot{q}_d)\hat{a} + Y_2(q_d, \dot{q}_d, \ddot{q}_d)\hat{a}_n \quad (2.10)$$

where \hat{a}_n contains the known parameters and \hat{a} contains the estimated parameters.

This control algorithm adapts to new situations because the estimated portion of the parameter vector is refined with each subsequent pass over a given trajectory. Slotine and Li have proposed an approach to AMBC that uses this parameter refinement approach to also compensate for controller limitations [25]. Their equation for the linear parameter vector is as follows [22]:

$$\hat{a} = \int \Gamma^{-1} Y^T(q, \dot{q}, \ddot{q}, \ddot{q}_r)[(\dot{q}_d - \dot{q}) + \Lambda(q_d - q)] \quad (2.11)$$

where

$$\dot{q}_r = \dot{q}_d + \Lambda(q_d - q) \quad (2.12)$$

$$\ddot{q}_r = \ddot{q}_d + \Lambda(\dot{q}_d - \dot{q}). \quad (2.13)$$

Γ^{-1} is a diagonal matrix, whose values control the adaption of the individual \hat{a} parameters; Λ is a ratio of the position to velocity feedback gains ($\Lambda = K_p/K_v$). The Slotine and Li algorithm has been successfully implemented, however, it has proven unreliable in the presence of velocity measurement noise [7]. Sadegh and Horowitz have proposed another method of AMBC implementation which they call "Desired Compensation Adaptation Law"[23]. Previous AFIT studies have validated the performance potential of this approach to Direct Adaptive Control [16, 22]. This implementation and study of AMBC algorithms has been the subject of on-going research at the Air Force Institute of Technology (AFIT).

2.2.1 AFIT Research

Research at the Air Force Institute of Technology has included the evaluation of Adaptive Model-Based Control (AMBC) algorithms on two very disparate ma-

nipulators. The same equations have been implemented on both platforms and have yielded roughly the same results - that is, both robots experienced increased tracking accuracy after successive passes over a single trajectory. Both of the implementations that will be discussed were performed using a digital computer. The delay inherent in a digital implementation is handled by using the error information from the previous sample time in the current cycle output torque calculations [15]. This causes the equations to take the following form:

$$\tau_{ff}(k) = Y_1[q_d(k), \dot{q}_d(k), \ddot{q}_d(k)]\hat{a}(k) + Y_2[q_d(k), \dot{q}_d(k), \ddot{q}_d(k)]\hat{a}_n(k) \quad (2.14)$$

$$\hat{a}(k) = \int_0^{T_s} \Gamma^{-1} Y_1^T(q_d(k), \dot{q}_d(k), \ddot{q}_d(k)) [\dot{e}(k-1) + \Lambda e(k-1)] \quad (2.15)$$

$$\dot{e}(k-1) = \dot{q}_d(k-1) - [q(k-1) - q(k-2)]/T_s \quad (2.16)$$

$$e(k-1) = q_d(k-1) - q(k-1) \quad (2.17)$$

$$\tau_{fb} = K_D \dot{e}(k-1) + K_P e(k-1) \quad (2.18)$$

where T_s is the sample period and the integration was accomplished using the Adams-Bashforth Two-Step method. The two implementations are discussed more in depth in the following subsections.

2.2.1.1 Implementation on a Vertically Articulated Robot

Leahy and Whalen have implemented an AMBC algorithm, of the type described above and gained insights into issues such as algorithm tuning, parameter initialization/convergence and asynchronous adaptation [16]. The manipulator used was a Unimate PUMA-560 vertically articulated robot. As described above, the algorithm they implemented requires only desired values for position and velocity to control movement of the robot. The control algorithm was then tested over a variety of trajectories, using both known and unknown payloads. This rigorous test scenario was designed to test the algorithm as completely as possible, over the full range of manipulator dynamics. The Leahy and Whalen findings indicate that the

AMBC control algorithm can adapt to a given trajectory after a minimal number of passes. In fact, peak trajectory tracking errors were reduced by a significant amount over classical control methods [16]. Figure 1.1 is simply shown to demonstrate the relative merits of a plain PD feedback controller versus an adaptive controller. The algorithm designated AMBC/H is the heuristically tuned algorithm developed by Leahy and Whalen. The designation AMBC/PBPA is the algorithm version that will be developed and discussed in later chapters. It can be seen that both versions of the adaptive feedforward controller clearly provide better tracking accuracy than the PD controller alone. Furthermore, they illustrate that both the AMBC/H and the AMBC/PBPA controllers provide essentially the same tracking accuracy.

One interesting point of this implementation was the method Leahy and Whalen used to determine the initial values of the linear parameter vector (\hat{a}). For this implementation, \hat{a} was partially composed of an estimate of system parameters. They note that giving different initial values to different parameters, can result in convergence to a minimal trajectory tracking error in different numbers of passes over that particular trajectory. Leahy and Whalen note that using the best estimate of the known physical values provides quicker tracking error convergence than does starting \hat{a} at zero.

Of more consequence to this study are Leahy and Whalen's findings concerning algorithm tuning. Through a heuristic, manpower intensive process they were able to fine tune the algorithm to optimal performance. This 'optimal' performance provided peak tracking error of typically less than 5 thousandths of a radian over the full test trajectory suite. The persons performing this fine tuning procedure were well versed in the dynamics of this manipulator - even so, the tuning process was performed gradually over the course of months. The method Leahy and Whalen used to tune the algorithm was largely an iterative approach, where individual elements of Γ^{-1} were incrementally changed. They state that due to the strong interdependence of the parameters, estimating them currently can be as much an art as a science.

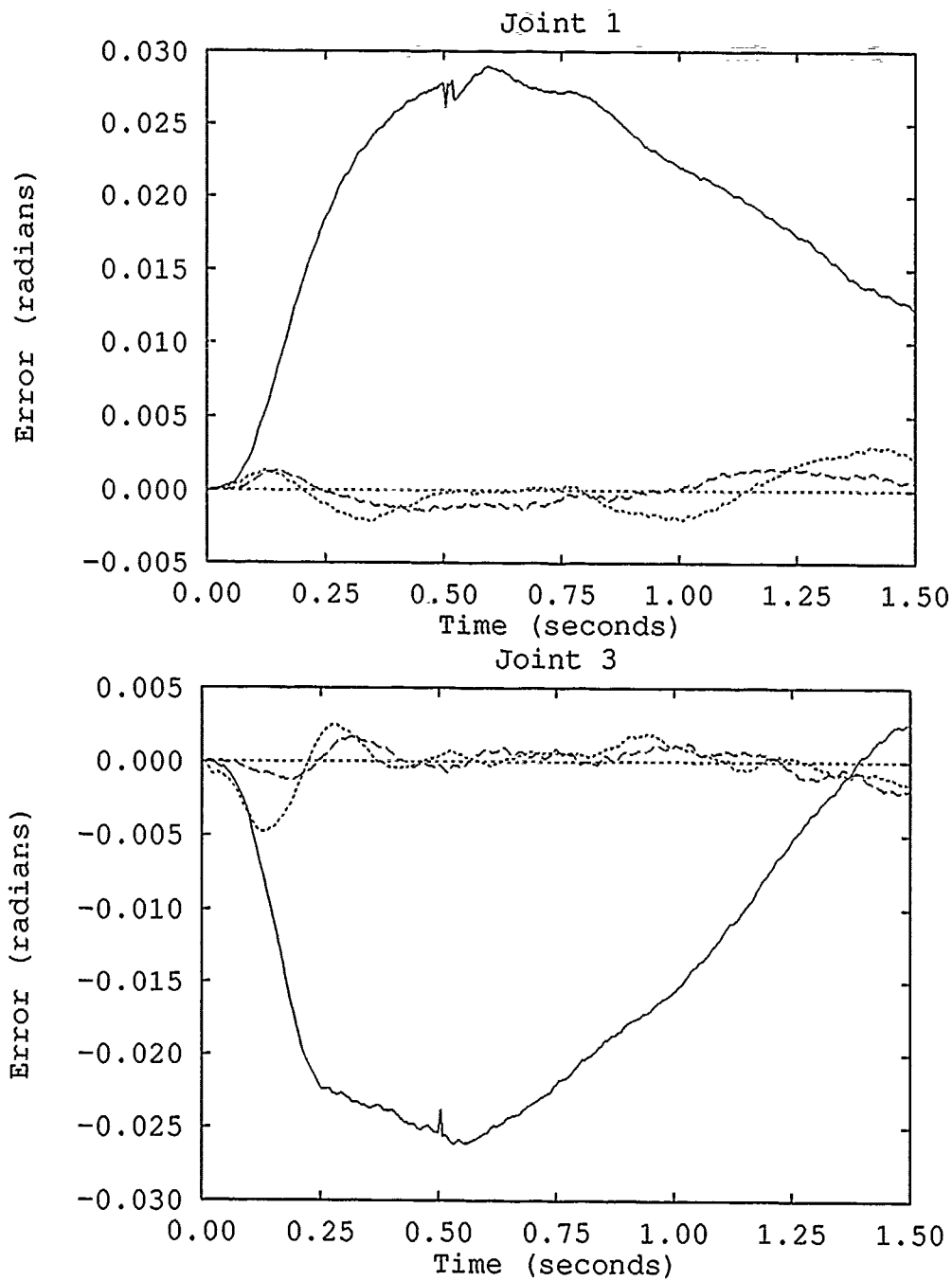


Figure 2.1. PD Only vs PD with AMBC - Trajectory 1

—	PD	- - - -	AMBC/H
.....	AMBC/PBPA		

Furthermore, Leahy and Whalen found that changing the Γ^{-1} values could result in either improved performance or disaster [16]. Figure 2.2 is supplied as an illustration of the effect of setting Γ^{-1} elements to a common value. While it is possible to track the given trajectory, peak tracking error increased dramatically over the case where a 'customized' set of Γ^{-1} values were used.

2.2.1.2 Implementation on a Dexterous Manipulator

A Master's Thesis, recently completed by Rainey, concerned the implementation and evaluation of an AMBC algorithm on a dexterous manipulator, configured for two degrees-of-freedom [22]. The robot used for this study was the Utah/MIT Dexterous Hand (UMDH). This robot is a tendon driven, multiple degree-of-freedom hand, developed primarily as a research tool for issues related to machine based artificial dexterity [11]. The UMDH is a prime candidate for a study of adaptive control algorithms because its internal dynamics are not as well known as the PUMA's. The implication of this is that any improvement in tracking accuracy will have been caused by the AMBC algorithm. Rainey used this robot to develop and test an AMBC algorithm in terms of suitability for human finger emulation [22]. As with the vertically articulated robot, AMBC provided a significant increase in tracking accuracy, as compared to classical control methods [22]. As did Leahy and Whalen, Rainey also used heuristic methods to fine tune his AMBC algorithm [16, 22].

2.3 Parameter Reduction

Once the parameters necessary for control of the robot have been determined, whether heuristically or mathematically, an appropriate regressor matrix can be determined such that $\tau = Y\hat{a}$. As stated earlier, in general, an arbitrary rigid body can be described with 10 dynamic parameters. When two rigid bodies are connected together, not all 20 parameters are needed to describe the entire system [1, 12, 8]. A number of papers have been published which deal with the issue of minimum param-

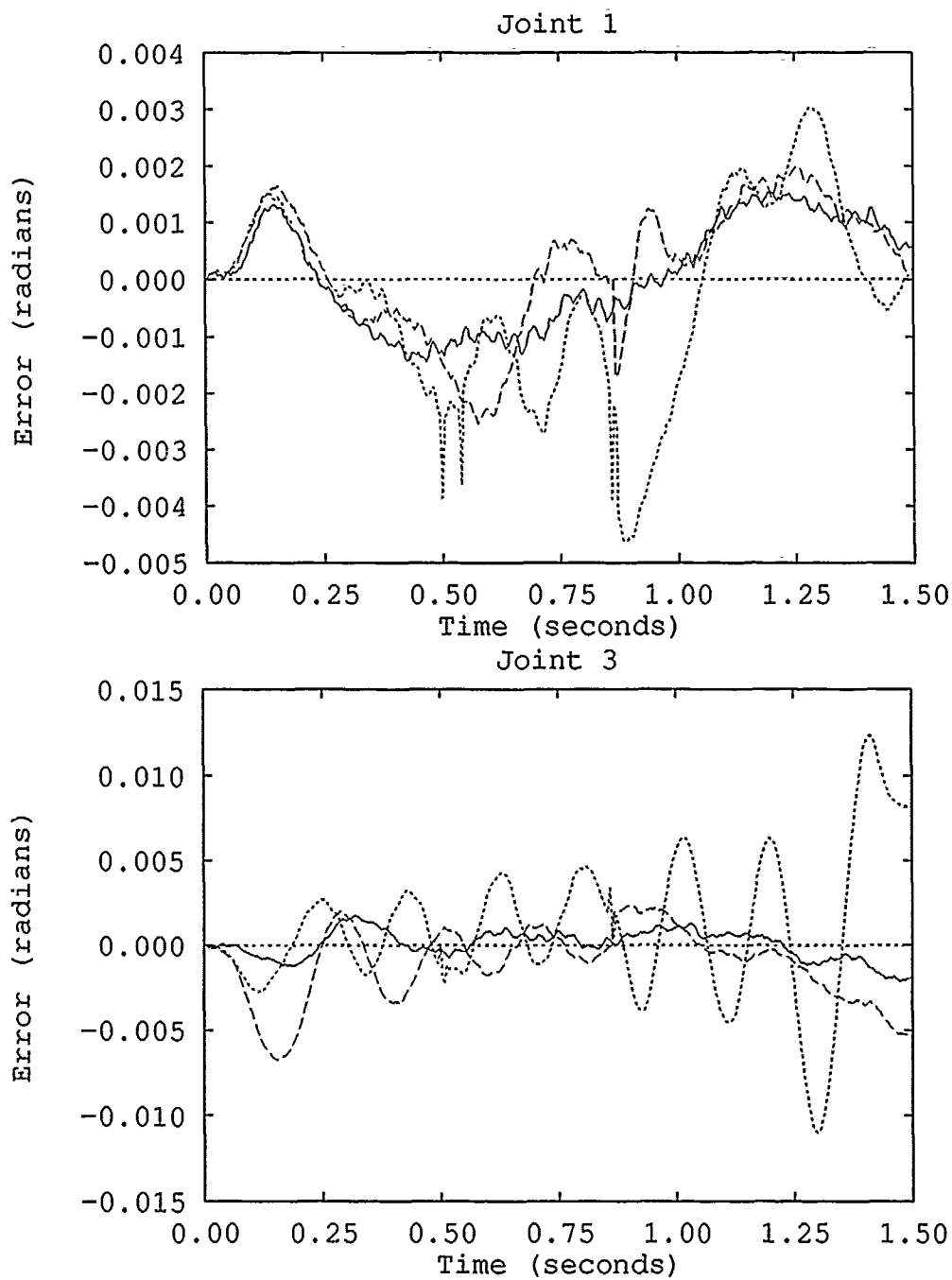


Figure 2.2. Effect of setting AMBC/H Γ^{-1} to a Common Value: Fifth Run, Trajectory 1

—	Customized	----	$\Gamma^{-1} = 25$
.....	$\Gamma^{-1} = 37.5$		

eter identification and parameter set reduction. These papers cover the spectrum from a basic method to simply quantify the number of necessary parameters [8] to another method to determine *which* parameters are needed [4], to a final method which determines not only which parameters are necessary, but also ranks them in order of sensitivity [5]. These three cases will be covered more extensively in the following sections.

2.3.1 Base Parameters of Manipulator Dynamic Models As stated above, while each link of a manipulator has ten associated parameters, when a robotic system of multiple links is considered, some parameters will be redundant. This is due to the fact that the relative motion of two adjacent links is restricted to one degree of freedom and the first link of the manipulator is typically connected to a fixed base by a joint [8]. A parameter set consisting strictly of non-redundant terms is commonly called the base parameter set for that particular manipulator. Determination of this base parameter set is useful in reducing the on-line calculations needed for accurate control of the robot. Mayeda et.al. have proposed a method for determining the minimum number of parameters for a system. The equation they arrived at is [8]:

$$p = 7N - 4\beta_1 \quad (2.19)$$

or, if the first joint axis is parallel to the gravity vector:

$$p = 7N - 4\beta_1 - 2 \quad (2.20)$$

where p is the total number of non-redundant parameters, N is the number of links in the manipulator and β_1 is the number of links connected by joints whose axes are always parallel to the first joint axis. This equation is useful for determining the quantity of parameters in the base parameter set; however, it will not determine

which parameters are important. This formula would be useful to a designer in verifying that he has reduced the parameter set to the correct size, but does not verify the accuracy of the reduction (e.g. a non-redundant parameter could possibly have been eliminated in favor of a redundant parameter).

2.3.2 Direct Calculation of the Minimum Set of Inertial Parameters

Gautier and Khalil have also presented a method for determining the minimal set of inertial parameters for serial robots [4]. Simply put, the procedure is to eliminate those parameters which have no effect and to regroup the remaining parameters as much as possible. To this end, Gautier and Khalil give a set of rules which permit the elimination of redundant parameters and qualifications for regrouping the remaining parameters. This same method is implemented by Gautier, using a numerical approach, in an earlier paper [3]. Either the Gautier numerical approach or the Gautier/Khalil rule based approach both show promise of being a straightforward procedure of identifying the base parameter set, but unfortunately do not provide any sort of element-to-element relationship in the parameter vector.

2.3.3 Principal Based Parameter Analysis

Even though the methods described above allow a AMBC algorithm designer to find the base parameter set, those methods do not address the relationship between the individual elements of the base parameter set. Ghodoussi and Nakamura have developed a mathematical method for determining the base parameter set, ranked in order of sensitivity, which they call Principal Based Parameter Analysis (PBPA). They present a systematic and general method to find those dynamic parameters that directly contribute to joint torques - that is, the base parameter set [5]. The result of this systematic analysis is a set of parameters numbered in order of sensitivity. Ghodoussi and Nakamura's theoretical development of PBPA is presented in the following sections.

2.3.3.1 PBPA Theoretical Development Each joint of a robot has ten associated primitive dynamic parameters, which describe how it is moved and positioned [20]. These parameters are the link mass m , the independent elements of the inertia tensor $I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$ and the three position elements of the mass centroid, r_x, r_y, r_z [5]. When moving the end effector into some arbitrary position, some of these parameters are more important than others and some parameters will be redundant. Through exhaustive testing, or with some insight into a particular manipulator, a reduced parameter set can be found. This reduced parameter set allows for the same total joint torque to be found, with less computation. The method presented by Ghodoussi and Nakamura not only identifies the minimal parameters specifically, but may also answer the more important question of how to fine tune an AMBC algorithm. The equations developed by Ghodoussi and Nakamura are presented in the following section.

2.3.4 Development of PBPA Equations The torque equations for a manipulator can be represented in the form [6]:

$$\tau = N(q, \dot{q}, \ddot{q})p \quad (2.21)$$

where p is the complete parameter vector and N is defined as:

$$N(q, \dot{q}, \ddot{q}) = \begin{pmatrix} n_1^T(q, \dot{q}, \ddot{q}) \\ \vdots \\ n_n^T(q, \dot{q}, \ddot{q}) \end{pmatrix}, \quad \text{where } n_i \in R^{N_{\text{parameters}}}. \quad (2.22)$$

The n subscript denotes total number of joints. n_i can be further defined as

$$n_i(q, \dot{q}, \ddot{q}) = n_o(q, \dot{q}, \ddot{q})K_i \quad (2.23)$$

Where n_0 is defined as a vector containing all the non-linear terms that appear in N and K_i is a constant matrix involving the kinematic parameters for link i .

N can now be represented as:

$$N \equiv \begin{pmatrix} n_o^T K_1^T \\ n_o^T K_2^T \\ \vdots \\ n_o^T K_m^T \end{pmatrix} \quad (2.24)$$

Next, a variation in torque due to a variation in the parameter set can be represented as:

$$\delta\tau = N(q, \dot{q}, \ddot{q})\delta p \quad (2.25)$$

and

$$J(\delta p) = \int \dots \int \|\delta\tau\|^2 dq_1 \dots dq_m d\dot{q}_1 \dots d\dot{q}_m d\ddot{q}_1 \dots d\ddot{q}_m \quad (2.26)$$

Combining the above equations, yields

$$J(\delta p) = \delta p^T \begin{pmatrix} K_1 & \dots & K_m \end{pmatrix} \begin{pmatrix} R & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & R \end{pmatrix} \begin{pmatrix} K_1^T \\ \vdots \\ K_m^T \end{pmatrix} \delta p \quad (2.27)$$

where

$$R \equiv \int_V n_o n_o^T dV \quad (2.28)$$

R is a symmetric, positive semidefinite matrix and is called the *Covariance Matrix of Nonlinearity*. A symmetric, positive semi-definite matrix can be shown in the form $R = \bar{R}\bar{R}$, where $R^{1/2} \equiv \bar{R}$. Substituting back into equation 2.27:

$$J(\delta p) = \delta p^T S^T S \delta p \quad (2.29)$$

where

$$S \equiv \begin{pmatrix} R^{1/2} K_1^T \\ R^{1/2} K_2^T \\ \vdots \\ R^{1/2} K_m^T \end{pmatrix} \quad (2.30)$$

Equation 2.30 implies that the nonlinear effect of q , \dot{q} and \ddot{q} on the contribution of the dynamic parameters to joint torques is taken into account via the covariance matrix of nonlinearity.

The Singular Value Decomposition of S is:

$$SVD(S) = U_s \Sigma_s V_s^T \quad (2.31)$$

and the Principal Base Parameters (denoted p^*) are simply found by:

$$p^* = V_s^T p = (P_1^* \cdots P_{N_p}^*)^T \quad (2.32)$$

$$p_i^* \equiv v_{s_i}^T p \quad (2.33)$$

The end result of PBPA is this reduced linear parameter vector p^* , which has an inherent relationship between its elements. This relationship will be exploited in the following chapter to develop a simple AMBC tuning procedure.

2.4 Conclusion

Adaptive Control will play a large part in the future of robotics. AMBC provides a tremendous improvement in tracking accuracy over plain PD controllers; however, the associated, heuristic tuning process has kept AMBC from transitioning from the laboratory environment to operational applications. Before AMBC can become widely used in everyday operations, some straightforward procedure for algorithm tuning must be found. Furthermore, this tuning procedure should not require a huge investment of manpower or extensive knowledge of the system dynamics. To date, no such procedure exists, even though at least one possible approach has been suggested. This study will explore the possibility of developing an AMBC algorithm and an associated procedure that details how to quickly and simply tune for optimal performance.

III. Adaptive Model-Based Control Development using Principal Base Parameter Analysis

3.1 Overview

Previous studies at the Air Force Institute of Technology have shown that Adaptive Model-Based Control (AMBC) algorithms can provide excellent tracking accuracy as compared to classical control methods [16, 22]. One of the reasons AMBC algorithms are not more widely used in operational systems is the method by which they must be fine tuned for optimal performance. This tuning process is currently heuristic and manpower intensive in nature. Before AMBC algorithms can gain wider acceptance outside of laboratory environments, the tuning process must mature considerably.

As discussed in the previous chapter, a method of parameter analysis recently proposed by Ghodoussi and Nakamura may provide a basis for a more acceptable tuning process. Their analysis method, Principal Base Parameter Analysis (PBPA), might be used in a process by which an AMBC algorithm can be tuned in minutes vice the current tuning time of hours or days.

This chapter will use PBPA to develop an AMBC algorithm which can be tuned with a single scaling factor. This development process will first be illustrated with a planar, two degree of freedom example. The procedure will then be applied to the PUMA-560, configured as a three degree of freedom platform (joints one, two and three actuated). Finally, the possibility of extending this analysis to a second manipulator, the Utah/MIT Dexterous Hand will be discussed.

3.2 AMBC Design with PBPA

Numerous forms of Adaptive Model-Based Control have been proposed and experimentally evaluated. The direct adaptive forms share the following general

format:

$$\tau = Y(q, \dot{q}, \ddot{q})\hat{a} + \tau_{fb} \quad (3.1)$$

$$\hat{a} = \int_0^{T_s} \Gamma^{-1} Y^T(q, \dot{q}, \ddot{q}) s \quad (3.2)$$

where τ_{fb} represents a linear feedback law and s is a vector of weighted tracking error. Two universal design considerations are the size and physical representation of the regressor (Y) and the method of tuning the adaptive gain matrix (Γ^{-1}). While multiple combinations of Y and \hat{a} can produce identical joint torques, different permutations of Y and \hat{a} result in variations in algorithm stability, computational complexity, and tracking performance [16]. The dependence of algorithm stability, parameter convergence, and tracking performance on the strength of the Γ^{-1} elements is also well known. As discussed in section 2.2.1.1, the current method of determining the strength of the individual Γ^{-1} elements is by trial and error. This heuristic tuning process is time consuming and provides no indication of the 'best' set of Γ^{-1} values. AMBC design using PBPA may provide a non-heuristic basis for tuning the Γ^{-1} elements. To illustrate this proposed design procedure, the linear parameter vector (\hat{a}), the regressor matrix and the adaptive gain matrix values are first determined for a 2 DOF ideal planar arm representation, based on links 2 and 3 of a PUMA-60. The extension of that procedure to a *real* 3 DOF PUMA configuration is then discussed. Finally, the foundation for extension of this process to the Utah/MIT Dexterous Hand is laid.

3.2.1 Ideal 2 DOF PUMA

Using standard Denavit-Hartenberg convention, the link parameter notation used by Spong and Vidyasagar, and for the moment neglecting drive system dynamics, the torque equations for joints 2 and 3 are written as [13, 26]:

$$\tau_2 = \ddot{q}_2(J_2 + J_3 + l_{c2}^2 m_2 + l_2^2 m_3 + l_{c3}^2 m_3 + 2l_2 l_{c3} m_3 \sin q_3)$$

$$\begin{aligned}
& + \ddot{q}_3(J_3 + l_{c3}^2 m_3 + l_2 l_{c3} m_3 \sin q_3) \\
& + 2\dot{q}_2 \dot{q}_3 l_2 l_{c3} m_3 \cos q_3 \\
& + \dot{q}_3^2 l_2 l_{c3} m_3 \cos q_3 \\
& - g l_{c2} m_2 \cos q_2 \\
& - g l_2 m_3 \cos q_2 \\
& - g l_{c3} m_3 \sin(q_2 + q_3)
\end{aligned}$$

$$\begin{aligned}
\tau_3 = & \ddot{q}_2(J_3 + l_{c3}^2 m_3 + l_2 l_{c3} m_3 \sin q_3) \\
& + \ddot{q}_3(J_3 + l_{c3}^2 m_3) \\
& + \dot{q}_2^2 l_2 l_{c3} m_3 \cos q_3 \\
& - g l_{c3} m_3 \sin(q_2 + q_3)
\end{aligned}$$

For this simple example, the nonlinear terms are grouped into the n_0 vector by inspection, yielding:

$$n_0^T = \begin{bmatrix} \ddot{q}_2 & \ddot{q}_2 \sin q_3 & \ddot{q}_3 & \ddot{q}_3 \sin q_3 & \cos q_3 \dot{q}_2 \dot{q}_3 \\ \cos q_3 \dot{q}_3^2 & \cos q_2 & \sin(q_2 + q_3) & \cos q_2 \dot{q}_2^2 \end{bmatrix} \quad (3.3)$$

with the corresponding vector of all primitive parameters given by:

$$p^T = \begin{bmatrix} J_2 & J_3 & m_2 l_{c2}^2 & m_3 l_2^2 & m_3 l_{c3}^2 \\ m_3 l_{c3} l_2 & m_2 l_{c2} g & m_3 l_2 g & m_3 l_{c3} g \end{bmatrix} \quad (3.4)$$

and the constant matrices K_i defined as:

$$K_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.5)$$

$$K_2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (3.6)$$

The Covariance Matrix of Nonlinearity is calculated by Equations 2.28, 3.3, and an appropriate set of integration limits on $|q|$, $|\dot{q}|$, and $|\ddot{q}|$. The values chosen for the integration limits are configuration specific. For this particular configuration, the values $|q_i| \leq \pi$, $|\dot{q}_i| \leq 2$, $|\ddot{q}_i| \leq 5$ were chosen. These particular values define the experimentally determined maximums for a PUMA robot [18]. Symmetrical limits, while not truly indicative of most industrial robots, significantly reduce the overall complexity of the PBPA. If non-symmetrical limits are used, the R matrix is less sparse, therefore complicating the results. Simplification via symmetrical integration limits becomes pivotal with higher degree of freedom cases (3 DOF and more). In fact, when the 3 DOF case was first investigated, real-world values were used for the integration [13]. The overall integration for non-symmetrical limits took almost 10 times longer than when symmetrical limits were used.

Forming S (Equation 2.30) and performing a Singular Value Decomposition yields V_s^T and therefore, the Principal Base Parameter Set, p^* . (Equations 2.31 and 2.32).

$$p^* = \begin{bmatrix} -0.2182(J_2 + l_{c2}^2 m_2 + l_2^2 m_3) - 0.6547(J_3 + l_{c3}^2 m_3) \\ -0.5345(J_2 + l_{c2}^2 m_2 + l_2^2 m_3) + 0.2673(J_3 + l_{c3}^2 m_3) \\ 0.02486(gl_{c2}m_2 + gl_2m_3) - 0.9994l_2l_{c3}m_3 \\ -0.7067(gl_{c2}m_2 + gl_2m_3) - 0.03516l_2l_{c3}m_3 \\ -gl_{c3}m_3 \end{bmatrix}$$

3.2.1.1 Formulation of the Adaptation Control Law

As discussed in section 3.2, the general form of a direct adaptive controller is $\tau = Y(q, \dot{q}, \ddot{q})\hat{a} + \tau_{fb}$. Feedforward torque is simply the product of the regressor matrix Y and a linear parameter vector \hat{a} . Since p^* has been defined as the set of non-redundant parameters, for this design procedure, set $\hat{a} = p^*$. For this design example, the next piece of the puzzle is to find Y . It is useful to explicitly point out that Y is not unique. The exact composition of Y depends on the system torque equations and the final make-up of the linear parameter vector.

The size of p^* implies that the regressor will be a 2 X 5 matrix (e.g. 2 joints X 5 p^* elements). The next step is to determine the actual regressor values. The terms in the regressor matrix can be determined using knowledge of the system torque equations and p^* , to solve a series of simultaneous equations. A symbolic mathematic manipulator, such as "Mathematica" [28] minimizes the drudgery. For the 2 DOF case, the system of equations was 9 equations with 9 unknowns. The value nine is simply the number of discrete, non-linear terms appearing in n_o^T . The size of n_o^T for the 3 DOF case implies a system of equations of 54 equations with 54 unknowns. This indicates that larger DOF manipulators will quickly become quite complex and their regressor increasingly more difficult to find - perhaps impossible to find without computer aid. It should be noted that finding the regressor matrix via simultaneous equations is straightforward, but time consuming. The result of the simultaneous equations for the 2 DOF equation is shown as Equation 3.7.

$$Y^T(q, \dot{q}, \ddot{q}) = \begin{bmatrix} -0.6547\ddot{q}_2 - 1.3093(\ddot{q}_2 + \ddot{q}_3) \\ -1.6036\ddot{q}_2 + 0.5345(\ddot{q}_2 + \ddot{q}_3) \\ -0.0498 \cos q_2 - 0.99938(2\dot{q}_2\dot{q}_3 \cos q_3 + \dot{q}_3^2 \cos q_3 + 2\ddot{q}_2 \sin q_3 + \ddot{q}_3 \sin q_3) \\ 1.4133 \cos q_2 + 0.03516(-2\dot{q}_2\dot{q}_3 \cos q_3 - \dot{q}_3^2 \cos q_3 - 2\ddot{q}_2 \sin q_3 - \ddot{q}_3 \sin q_3) \\ \sin(q_2 + q_3) \end{bmatrix} \begin{bmatrix} -1.3093(\ddot{q}_2 + \ddot{q}_3) \\ 0.5345(\ddot{q}_2 + \ddot{q}_3) \\ -0.99938(\dot{q}_2^2 \cos q_3 + \ddot{q}_2 \sin q_3) \\ -0.03516(\dot{q}_2^2 \cos q_3 + \ddot{q}_2 \sin q_3) \\ \sin(q_2 + q_3) \end{bmatrix} \quad (3.7)$$

3.2.1.2 *Tuning the AMBC Algorithm*

Γ^{-1} is a diagonal adaptive gain matrix whose values control the rate of adaption of the individual \hat{a} parameters. As discussed in section 2.3.3.1, Principal Base Parameter Analysis not only eliminates all redundant parameters yielding the base parameter set, but also ranks the base parameter set elements in order of sensitivity. This relative ordering of the individual parameter set elements is what provides the basis for non-heuristic tuning of the AMBC algorithm. The elements of \hat{a} (post PBPA) have an inherent relationship to one another, and there is an element to element correspondence between \hat{a} and Γ^{-1} . Therefore, Γ^{-1} should possess the same type of relationship among its elements.

The physical values for the inertial parameters (e.g. link lengths, masses, etc.) are now substituted into \hat{a} . The resultant is a numerical vector and is used as the diagonal base value for Γ^{-1} . Assuming that the magnitudes of the individual parameters are a valid indication of their relative impact on controller torque, and therefore tracking performance, tuning is reduced to varying the entire Γ^{-1} matrix by a scaling factor, as opposed to tuning individual elements. Determining the validity of that assumption is the objective of the experimental analysis. However, before this analysis can take place, p^* and the associated regressor for the three DOF PUMA must be determined.

3.2.2 *The Three DOF Case*

In order to conduct a realistic evaluation of incorporating PBPA into AMBC design, the two DOF procedure was extended to the three positioning links of the PUMA. The lengths of the equations make their inclusion at this point in the paper unrealistic, however, they can be found in the appendices. The starting point was to determine the primitive parameter vector that represented all the classical dynamics of these three links of the manipulator. As for the two DOF case, the first step in this design procedure was to take the torque equations and re-arrange them into the

form shown in Equation 2.21. This re-arrangement resulted in a primitive parameter vector (n_o^T) of all of the dynamic parameters, but excluding friction. Following the same steps as detailed for the two DOF case, PBPA produced an 18 element principal base parameter vector (p^*). A realistic dynamical representation of the PUMA must also incorporate drive system information. Previous studies indicate that inclusion of coulombic and viscous friction torques in each joint is sufficient [18, 24]. Since the linear coefficients of those forces are independent of inertial dynamics, they can simply be appended onto the nominal p^* vector. Including them in the primitive parameter vector would only have complicated the PBPA and would have contributed no additional information. Initial attempts at PBPA for the three DOF case included the friction terms in the primitive parameter vector and they remained intact (e.g. they were not reduced further) by PBPA. All of this considered, the six friction parameters increases the size of the complete principal base parameter vector to 24 elements.

Once the complete p^* vector was found, physical values for the variables (e.g. masses, link lengths, experimentally determined friction coefficients, etc.) were substituted and the individual p^* elements were re-arranged in decreasing order of their absolute magnitude. p^* was next normalized to the magnitude of its smallest element. The final product was a linear parameter vector, consisting of 24 elements, made up of the minimal required inertial parameters, as well as the viscous and coulombic friction parameters, arranged in decreasing order of relative absolute magnitude and normalized to the smallest relative value. This normalized p^* , rounded to the nearest whole number, is 8530, 2896, 2254, 2043, 1782, 1348, 1171, 1048, 1048, 448, 396, 386, 329, 204, 185, 137, 99, 71, 64, 35, 9, 4, 3, 1. In the actual implementation, greater precision was used - the rounding done here was applied simply to illustrate the relative values.

Finally, as described in the previous section, the regressor was found using the initial torque equations and a system of simultaneous equations, solved via Mathe-

matica on a Sparc2 workstation. Using a comparable machine and given the torque equations and integration limits, a user experienced in both PBPA and Mathematica could arrive at a final p^* , for a typical three DOF manipulator, in under one day. This process is somewhat time-consuming, however, the inconvenience should be traded off against the relative resultant ease of tuning after implementation.

3.3 The Utah/MIT Dexterous Hand Implementation

Using the same torque equations as Rainey, and following the design steps laid out above, PBPA was done on the UMDH, configured as a two degree of freedom manipulator. Furthermore, the results of the PBPA were incorporated into an AMBC algorithm; however, work was halted just short of implementation. As discussed in chapter 2, there was significant effort to be expended on making the UMDH operational, before the AMBC/PBPA experimental analysis could be done. Unfortunately, this extra effort caused this part of the research to be aborted. At time of publication, the host computer for the ARCADE software was only able to achieve rudimentary communication with the UMDH. This interface was not enough to implement and evaluate any sort of advanced control algorithms. Consequently, no analysis of results for the UMDH will be discussed in chapter 4.

3.4 Summary

A procedure for the design of a non-heuristically tuneable adaptive controller² was developed by way of a two degree of freedom planar example. This two DOF example, based on joints two and three of the PUMA 560 used the original torque equations to perform Principal Base Parameter Analysis. The resultant principal base parameter vector was then used to find an AMBC regressor matrix and corresponding base Γ^{-1} values. Finally, this two DOF example was extended to the three positioning links of the PUMA 560. Some of the pitfalls of the approach (i.e. selection of integration limits, exponential growth of the simultaneous equation) were

also discussed. The next logical step in this study is to actually implement the 3 DOF AMBC/PBPA controller and contrast its performance against the performance achieved with the heuristically reduced and tuned controller.

IV. PBPA Test and Analysis

4.1 Overview

The true test of the potential of an advanced control algorithm can only be determined via experimental test and evaluation. To that end, the AMBC/PBPA algorithm has been exhaustively tested and the results analyzed. The goal of these experimental evaluations was to validate concepts, not to produce the optimal PUMA-specific algorithm. Therefore all test runs produced using the AMBC/PBPA algorithm had the same, single Γ^{-1} scaling factor. No attempt was made to 'fine-tune' the AMBC/PBPA algorithm further. This restraint from fine tuning is important because these results are to be compared against the AMBC/H algorithm which was fine tuned over the course of months. The initial thrust of this chapter is to describe the test environment and algorithm initialization issues, followed by a description of the test trajectories. The test suite follows that made in past AFIT studies [14, 15, 16, 17, 18]. The AMBC/PBPA algorithm will be compared and contrasted against the AMBC/H algorithm in areas such as simple trajectory tracking, robustness to payload variation, parameter convergence, pattern learning and sensitivity to reduction of parameters. The two algorithms will be compared numerically, as well as graphically. In this chapter, only representative plots will be shown - the comprehensive collection of plots can be found in the appendices.

4.2 Test Environment

Principal Base Parameter Analysis is an avenue that can be used by an AMBC designer to come up with an easily tuned control algorithm. However, does this algorithm provide adequate tracking accuracy? This experimental evaluation will compare the AMBC/PBPA algorithm against the AMBC/H algorithm, as used in previous AFIT studies. The ultimate goal is to determine if PBPA provides a sound basis for tuning the adaptation gain matrix of an AMBC algorithm with a single

scaling factor. Furthermore, it is to show the ability to reduce the required set of parameters needed by an adaptive algorithm, without introducing significant additional tracking errors.

The algorithm chosen to accomplish this is the Sadegh and Horowitz version of an adaptive controller, which has been used in previous AFIT studies [23, 22, 16]. This Desired Compensation Adaptation Law (DCAL) was discussed in depth in section 2.2.1. The PD gains were set to stiff values employed in previous model based studies [18, 16] and reiterated in table 4.1.

Table 4.1. Stiff PD Feedback Gains

	Joint 1	Joint 2	Joint 3
K_p	640.0	1330.0	360.0
K_v	72.0	130.0	25.0

The same form of the DCAL was used for both cases; that is, for the heuristically tuned case (ala Leahy and Whalen) and for the non-heuristically tuned case (this study). While the same same form is used (e.g. $\tau_{ff} = Y\hat{a}$), it should be noted that both Y and \hat{a} will be composed of entirely different values. As described in section 2.2.1.1 these two cases will be referred to as 'AMBC/H' and 'AMBC/PBPA', respectively (AMBC/H meaning AMBC with heuristic tuning). Finally, these experimental evaluations were conducted at a servo rate of 222 Hertz, using the AFIT Robotic Control Algorithm Development and Evaluation (ARCADE) environment [18].

4.2.1 Description of Trajectories The test trajectories chosen correspond to those used by Leahy and Whalen for their analysis of a Direct Adaptive Controller for Industrial Manipulators [16]. Their results will be used as a benchmark for comparison. The chosen trajectories, listed in Table 4.2, were first performed under zero payload conditions. Next, to prove adaptability under loaded conditions, two of

the trajectories were also run with a 2 Kg brass disk attached to the link six mounting flange [16]. Motor saturation constraints limited payload testing to Trajectories 1 and 3 only.

Table 4.2. Test Trajectories (degrees)[16]

Number	Start	Finish	Time (sec)
1	-50,-135,135	45,-90,30	1.5
2	-50,-205,90	45,-160,-15	2.0
3	0,-180,180	95,-135,75	2.0
4	0,-90,90	-95,-135,-15	2.0
5	0,-45,135	-95,-90,20	2.0
6	0,-90,90	95,-135,195	1.5

Trajectories 1, 2, and 3 each have an angular movement of $(95^\circ, 45^\circ, -105^\circ)$. Trajectory 1 and Trajectory 6 both travel the trajectory in 1.5 seconds while Trajectories 2 - 5 are performed in 2.0 seconds. The desired trajectory velocity and acceleration components will be identical for Trajectories 2 and 3. Trajectories 4 and 5 apply identically generated trajectory commands to different initial conditions. When compared to Trajectories 2 and 3, the initial positions differ and the movement of Joints 1 and 2 is opposite. (Trajectories 4 and 5 move $(-95^\circ, -45^\circ, -105^\circ)$). The respective desired position and acceleration terms should also differ in sign from those of Trajectories 2 and 3. These two test trajectories permit testing to consider the effects of different starting positions and direction of motion.

The movement of Trajectory 6 is similar to Trajectory 1. Joint 3 moves in the opposite direction while Joint 1 and 2 movement is the same. Desired velocity and acceleration terms for Joint 3 should have opposite signs. This trajectory can be used to determine the effects of different starting positions, direction of motion, and trajectory speed [24]. This selection of trajectories will allow evaluation of the effects of different starting positions and trajectory speeds.

4.2.2 Initialization of \hat{a}

To be consistent with Leahy and Whalen's testing, only 16 estimated parameters were used by the DCAL. While Leahy and Whalen chose 16 parameters based on payload sensitivity analysis, the 16 parameters chosen for the AMBC/PBPA algorithm were simply the 16 most sensitive parameters (as indicated by the Principal Base Parameter Analysis). The remaining, non-estimated parameters were given their nominal physical values. Leahy and Whalen have determined that 16 parameters rivaled the tracking performance of their robust feedback algorithm, while expanding the \hat{a} vector beyond 16 elements produced a negligible effect on tracking accuracy [16]. This inference is consistent with AMBC/PBPA - referring to the values given for the normalized p^* (section 3.2.2), it can be seen that $p^*(17)$ is approximately two orders of magnitude less than $p^*(1)$. Consequently, the inclusion of parameters 17-24 in the adaptation process would provide a very small additional feedforward torque in comparison to parameters 1-16.

One other point should be explicitly made concerning the non-estimated parameters. To reiterate, the total size of the AMBC/H parameter vector is 28 elements; total size of the AMBC/PBPA parameter vector is 24 elements (refer to section 3.2.2). This implies that there are 12 and 8 non-estimated parameters for the AMBC/H and the AMBC/PBPA algorithms, respectively. The non-estimated elements of each parameter vector are given their nominal physical values and remain constant throughout the entire trajectory (e.g. do not adapt). These non-estimated parameters contribute to the feedforward torque as shown in Equation 2-10.

4.2.2.1 AMBC/H Initialization of \hat{a}

For all cases where the AMBC/H algorithm was utilized, \hat{a} was initialized to its approximate physical values, as given by Tarn and Bejczy [27]. This initialization of \hat{a} causes the trajectory error to be significantly lower in the first run, as opposed to initially setting \hat{a} to zero [16]. These physical values are used simply as an approxi-

mation to what the values should be; the adaptive controller will fine-tune them to more appropriate values during the course of the adaption.

4.2.2.2 AMBC/PBPA Initialization of \hat{a}

For the cases where the AMBC/PBPA algorithm was used, \hat{a} was initialized to a pre-determined set of values. This pre-determined set of values was found by running Trajectory 2 until steady state error occurred. This particular set of values was chosen because Trajectory 2 was arbitrarily decided to be the most 'benign' of all the trajectories. Consequently, initial runs on this trajectory were less likely to become unstable while the algorithm was tuned (via the adaptive gain matrix scale factor). The resultant \hat{a} values were then used as the initial \hat{a} values on all other trajectories (Table 4.3). At first, the AMBC/PBPA \hat{a} initialization was attempted in the same manner as the AMBC/H \hat{a} initialization - that is, the values were set to their approximate physical values. However, using this approach, the PUMA was unable to successfully track a subset of the test trajectories. This seems to indicate that one (or more) of the approximations used is (are) grossly out of line.

4.2.3 Γ^{-1} Values

A prime advantage of PBPA can now be shown. Referring to Equation 2.15, Γ^{-1} is a diagonal matrix, used to tune the relative contribution of each individual element of the linear parameter vector to the overall feedforward torque. Leahy and Whalen note that with their previous method of reducing the parameter vector through experimentation, the Γ^{-1} matrix must also be tuned heuristically [16]. As discussed in section 2.2.1.1, while it is possible with the AMBC/H algorithm to simply set each element of Γ^{-1} to a common value, if the value chosen is too large, tracking accuracy suffers, if the value chosen is too small, adaptation time increases. The Leahy/Whalen method was to iteratively tune each element, seeking best overall performance. Leahy and Whalen were able to achieve admirable tracking accuracy, but at the expense of tuning the algorithm over the course of months.

With AMBC/PBPA, the adaptive gain matrix (section 3.2.2) can be tuned as a whole because each element is relative to the others. To change the rate of adaptation of the system, the entire AMBC/PBPA Γ^{-1} would simply need to be multiplied by a different scaling factor; however, the AMBC/H Γ^{-1} would need to be manually re-tuned, element by element. For all testing, the Γ^{-1} values used for the AMBC/H algorithm are (120,120,120,0,90,90,90,15,150,5,80,30,15,80,80,80) [16]. The Γ^{-1} values for the AMBC/PBPA version of the algorithm are the normalized values of the p^* vector (section 3.2.2), scaled by the multiple 0.020. The value 0.020 was chosen because it was the maximum scaling factor with which tracking was possible over all six trajectories. Incidentally, approximately 1 hour was consumed settling on this particular scale factor. The time spent tuning the AMBC/PBPA algorithm compares favorably to the months spent tuning the AMBC/H algorithm. If it were desired to increase the amount of feedforward torque to the system (therefore decreasing the time to steady state error), it would simply be a matter of increasing the single scaling factor; however, it should be noted that increasing the scale factor may cause instability on one or more of the test trajectories.

4.3 Analysis of Data

4.3.1 Evaluation Over Standard Trajectories This series of figures (Figures 4.1-4.5) demonstrates the performance of both algorithms showing selected trajectories and joints, no payload. Each figure shows the first and fifth pass of each version of the adaptive controller over the noted trajectory. Most of the figures have a common thread: the AMBC/PBPA algorithm starts with a slightly larger tracking error and still has not caught the AMBC/H algorithm by the fifth pass. In fact, this trend continues; that is, the AMBC/H algorithm consistently shows a tracking error somewhat less than the AMBC/PBPA algorithm (Figure 4.6); furthermore, the performance of the two algorithms never converge. However, it should be noted that the difference in tracking error is normally in the area of a few thousandths of

a radians. Regardless, the AMBC/PBPA algorithm essentially reaches steady state error after approximately five passes (as does the AMBC/H algorithm).

4.3.2 Adaption of \hat{a}

Table 4.3 shows the final values of the AMBC/PBPA \hat{a} for selected trajectories. They are seen to be quite different from their best estimate physical values; however, the steady state values are in the same realm over completely different trajectories. This is the one area where heuristics have not been completely eliminated. As described in section 4.2.2.2, the initial \hat{a} used for all trajectories was the set of values found after the algorithm reached steady state error on Trajectory 2. One heuristic way to arrive at an initial set of \hat{a} values was to switch the magnitudes of parameters one and two. This method usually allowed the manipulator to track the trajectory without becoming unstable, but not always. This particular issue will be addressed further in chapter 5.

4.3.3 Robustness to Payload Variation The AMBC/PBPA algorithm provides essentially the same robustness to payload variation as does the AMBC/H algorithm (Figure 4.7 and 4.8). These figures show the performance of both the AMBC/H and the AMBC/PBPA algorithms over Trajectories 1 and 3, carrying a 2 Kg payload. The performance of both is degraded, as compared to their no load counterparts; however, both the AMBC/H algorithm and the AMBC/PBPA algorithm provide essentially the same accuracy after five passes.

4.3.4 Decreasing Number of Estimated Parameters

Since a byproduct of the Principal Base Parameter Analysis is an ordering of the parameters by strength, there is now a mathematical basis for decreasing the number of estimated parameters. Leahy and Whalen used their extensive knowledge of the PUMA dynamics as a basis for parameter reduction. Unfortunately, most industrial applications will not have the luxury of that expertise. Since the elements of p^* are

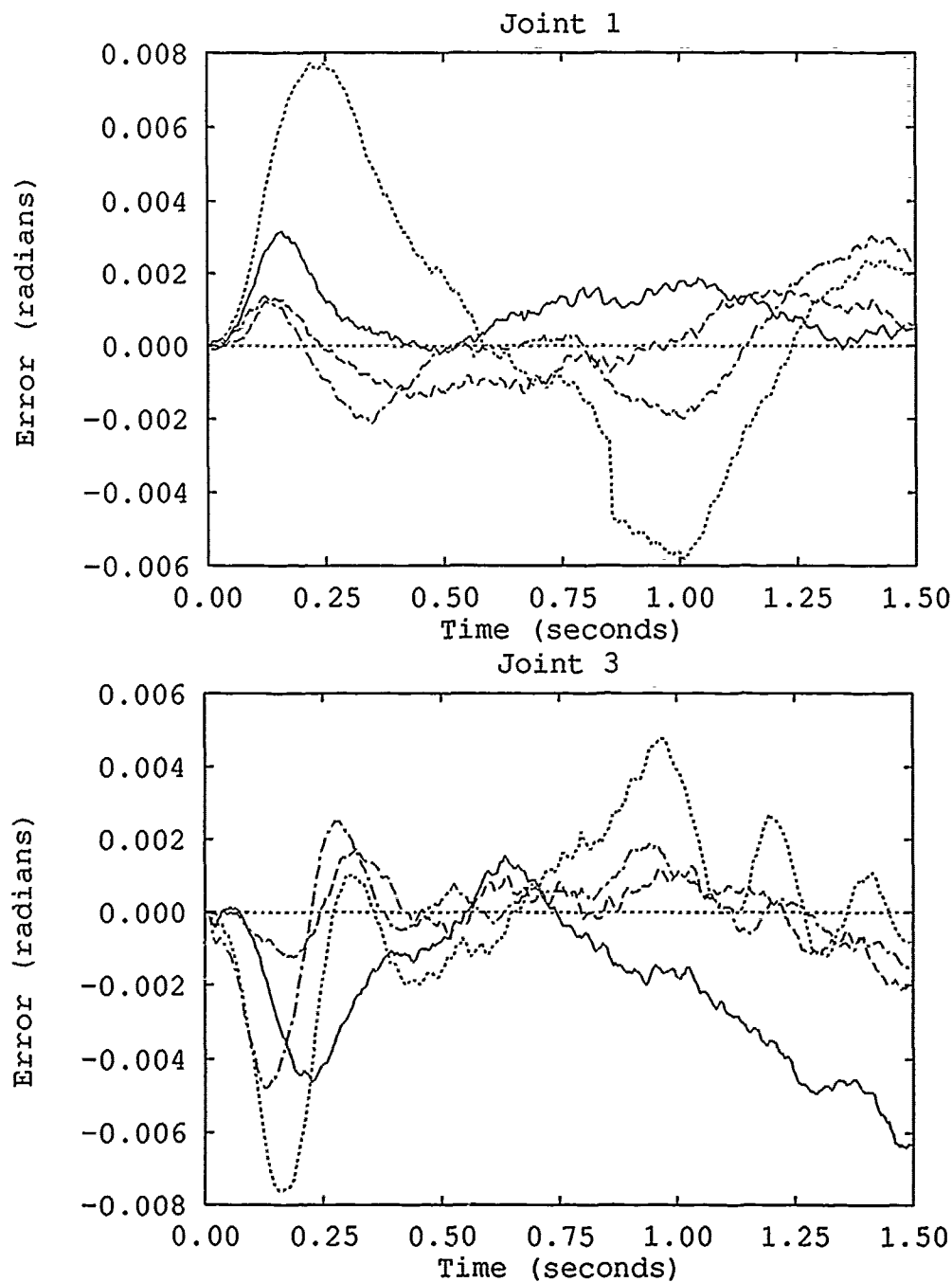


Figure 4.1. AMBC/H vs. AMBC/PBPA - Trajectory 1

—	AMBC/H - Run 1	AMBC/PBPA - Run 1
- - -	AMBC/H - Run 5	- . - .	AMBC/PBPA - Run 5

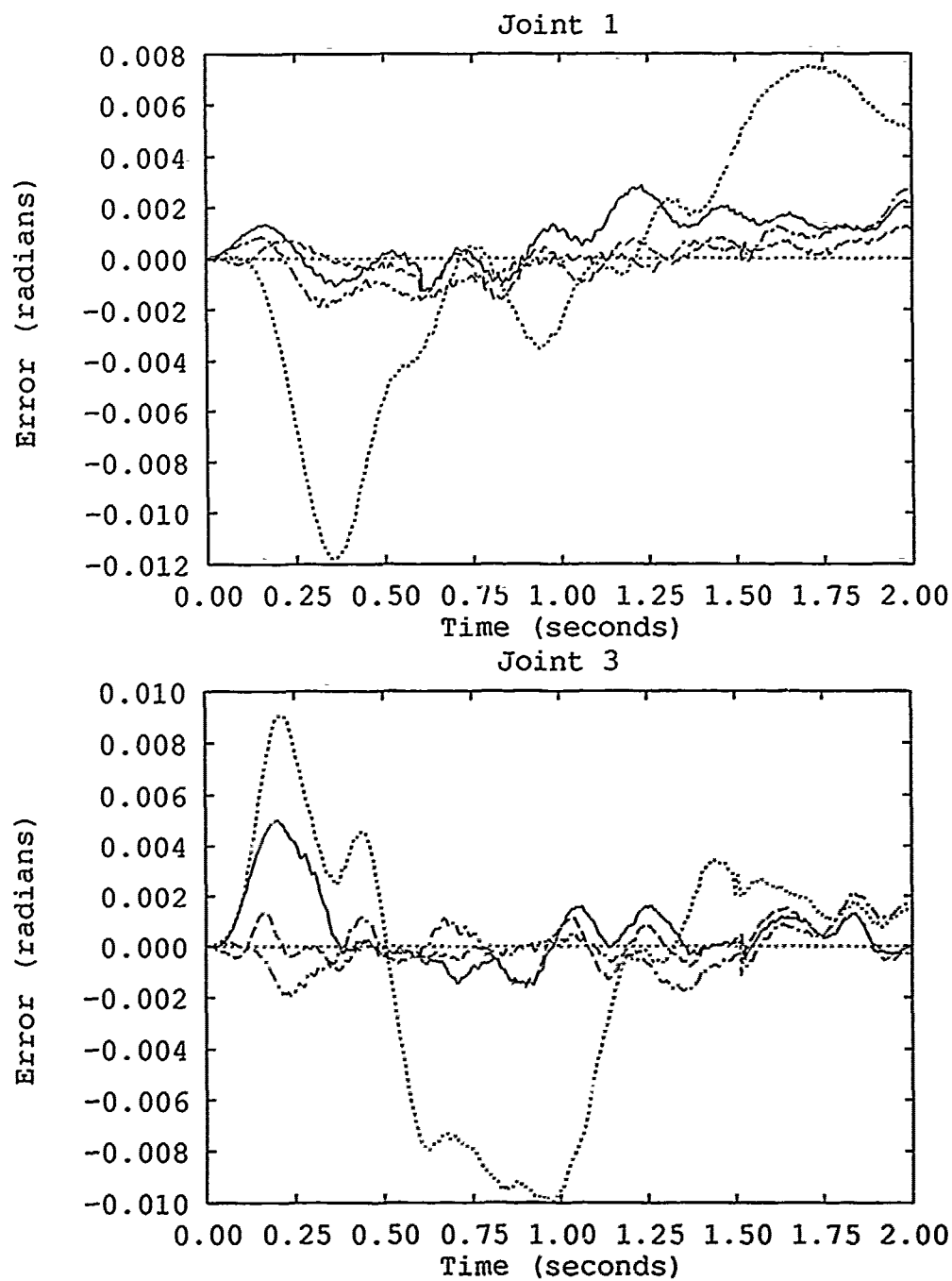


Figure 4.2. AMBC/H vs. AMBC/PBPA - Trajectory 2

—	AMBC/H - Run 1	AMBC/PBPA - Run 1
- - -	AMBC/H - Run 5	- . - .	AMBC/PBPA - Run 5

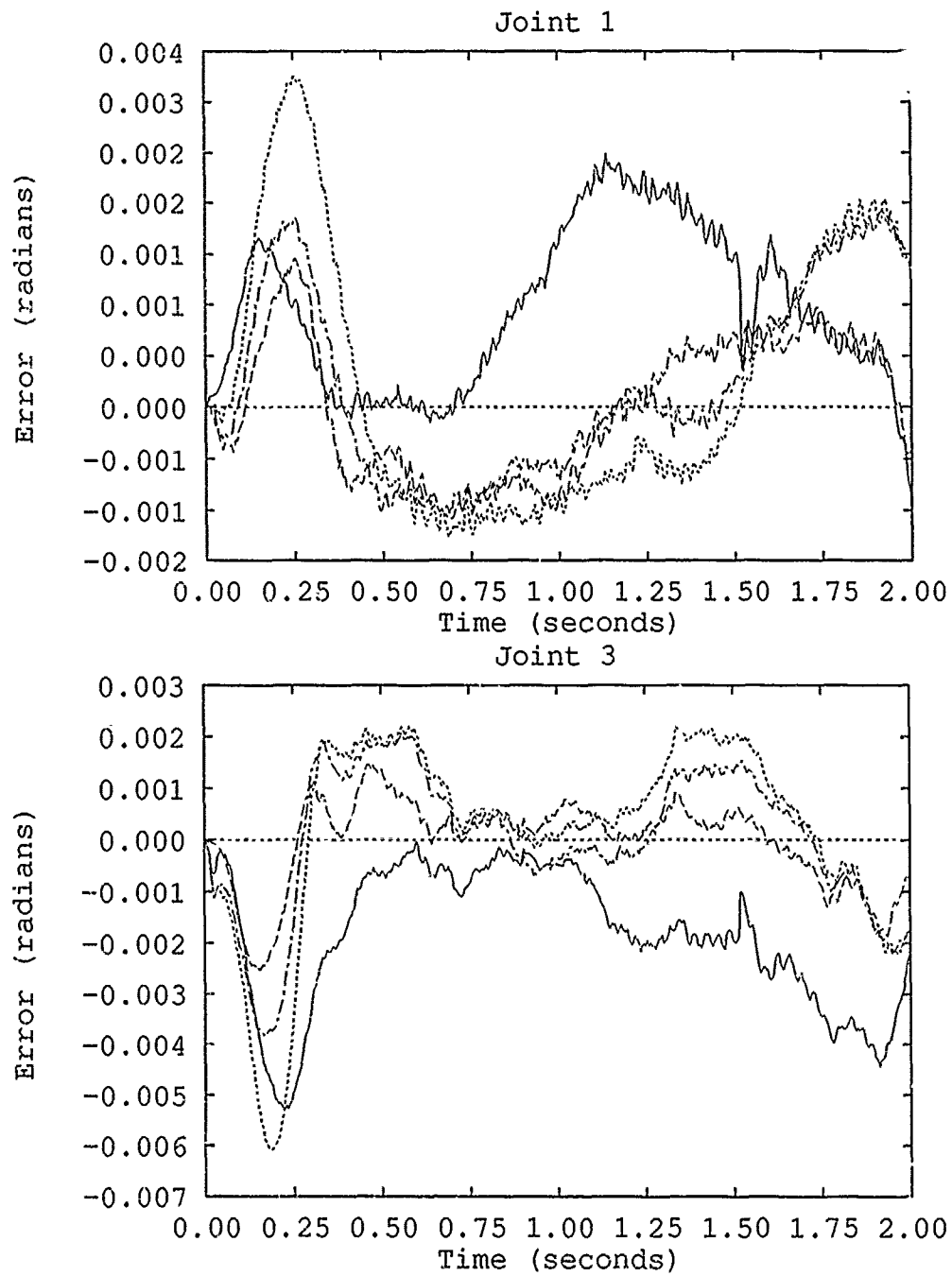


Figure 4.3. AMBC/H vs. AMBC/PBPA - Trajectory 3

—	AMBC/H - Run 1	AMBC/PBPA - Run 1
- - -	AMBC/H - Run 5	- · - · -	AMBC/PBPA - Run 5

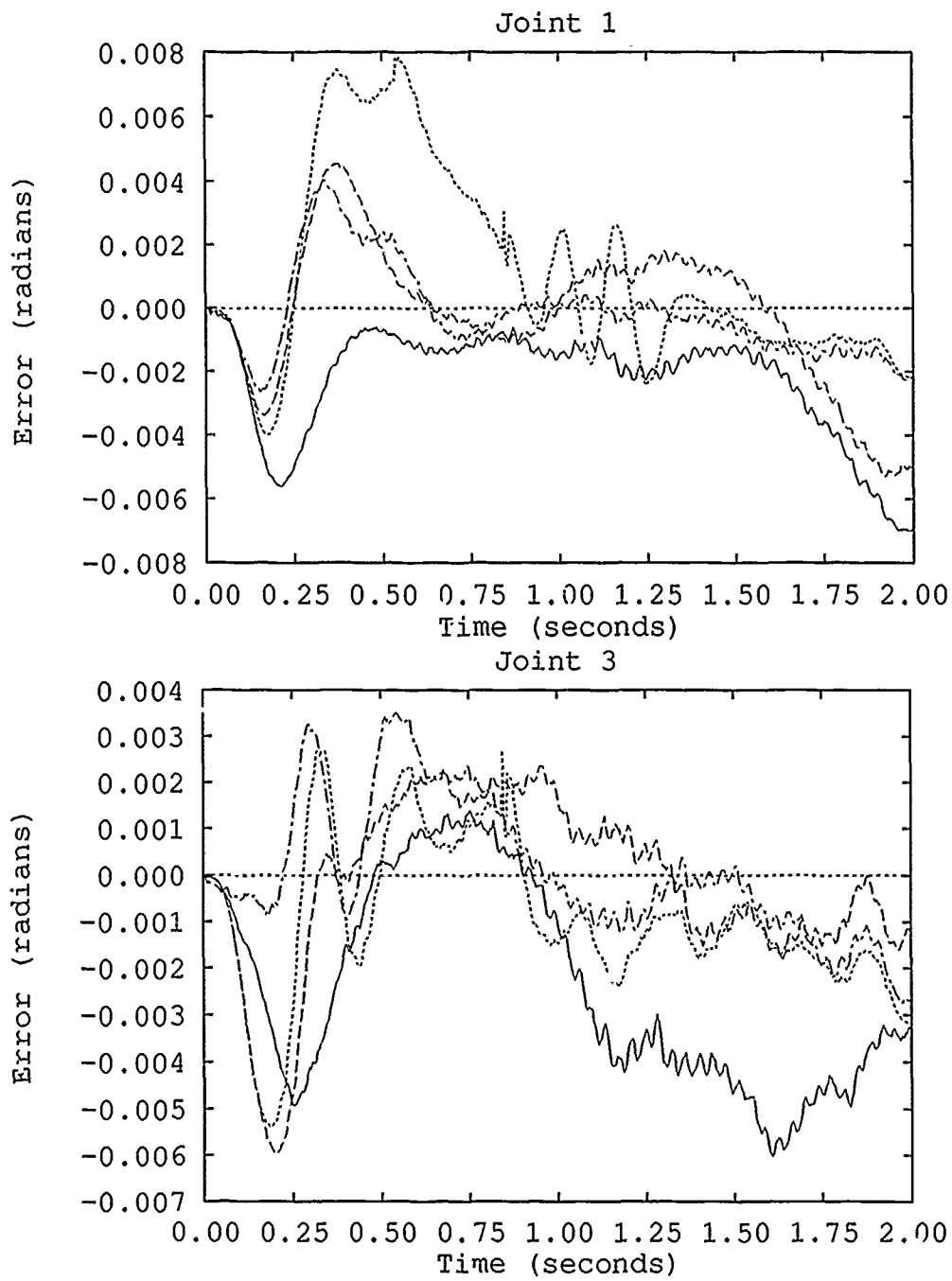


Figure 4.4. AMBC/H vs. AMBC/PBPA - Trajectory 5

—	AMBC/H - Run 1	AMBC/PBPA - Run 1
- - -	AMBC/H - Run 5	- . - .	AMBC/PBPA - Run 5

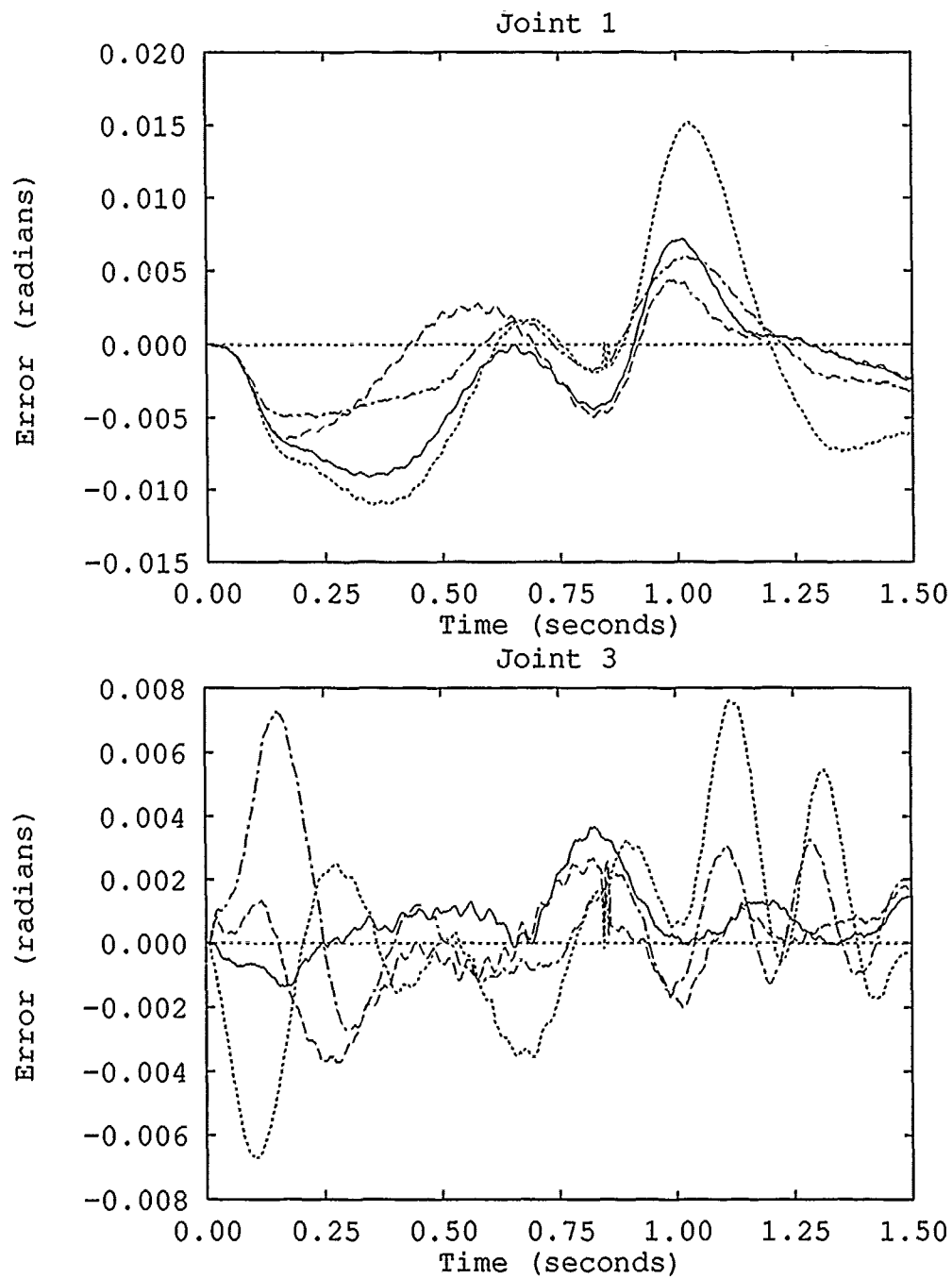


Figure 4.5. AMBC/H vs. AMBC/PBPA - Trajectory 6

—	AMBC/H - Run 1	AMBC/PBPA - Run 1
- - -	AMBC/H - Run 5	- · - · -	AMBC/PBPA - Run 5

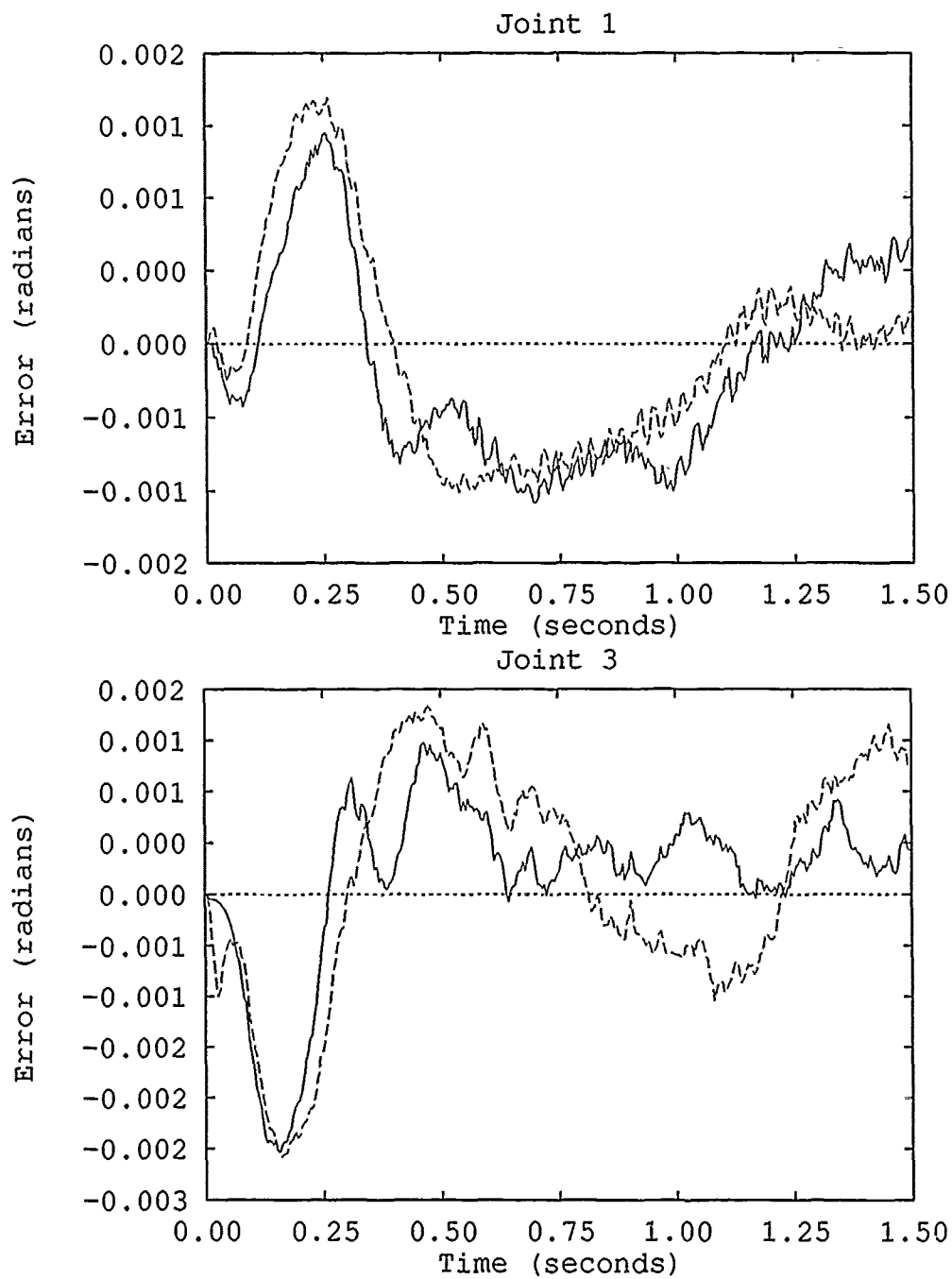


Figure 4.6. Illustration that AMBC/H and AMBC/PBPA do not converge - Trajectory 3

—	AMBC/H: Run 5	----	AMBC/PBPA: Run 10
---	---------------	------	-------------------

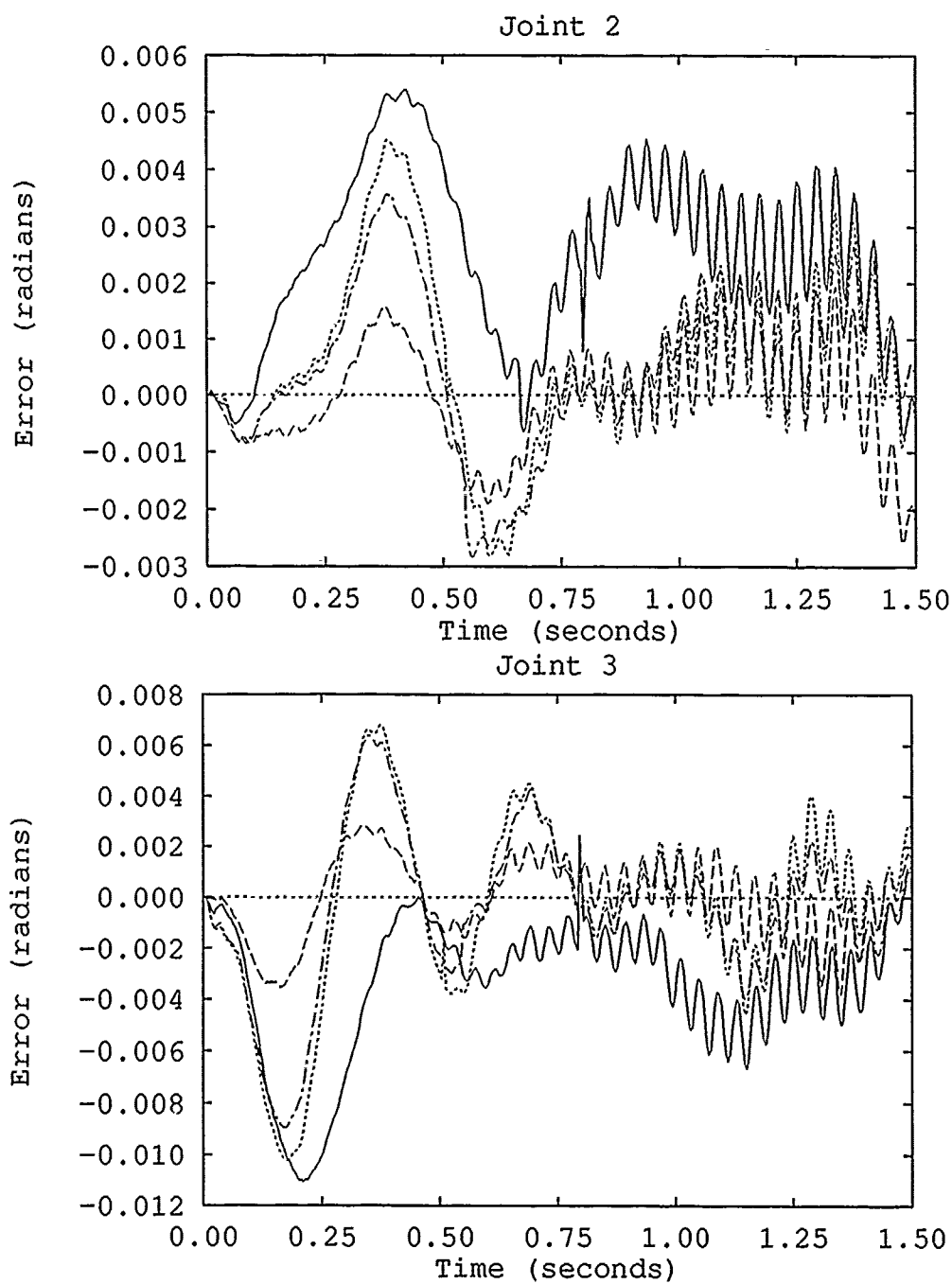


Figure 4.7. AMBC/H vs. AMBC/PBPA - Trajectory 1 with 2KG Payload

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
.....	AMBC/PBPA - Run 1
- . - .	AMBC/PBPA - Run 5

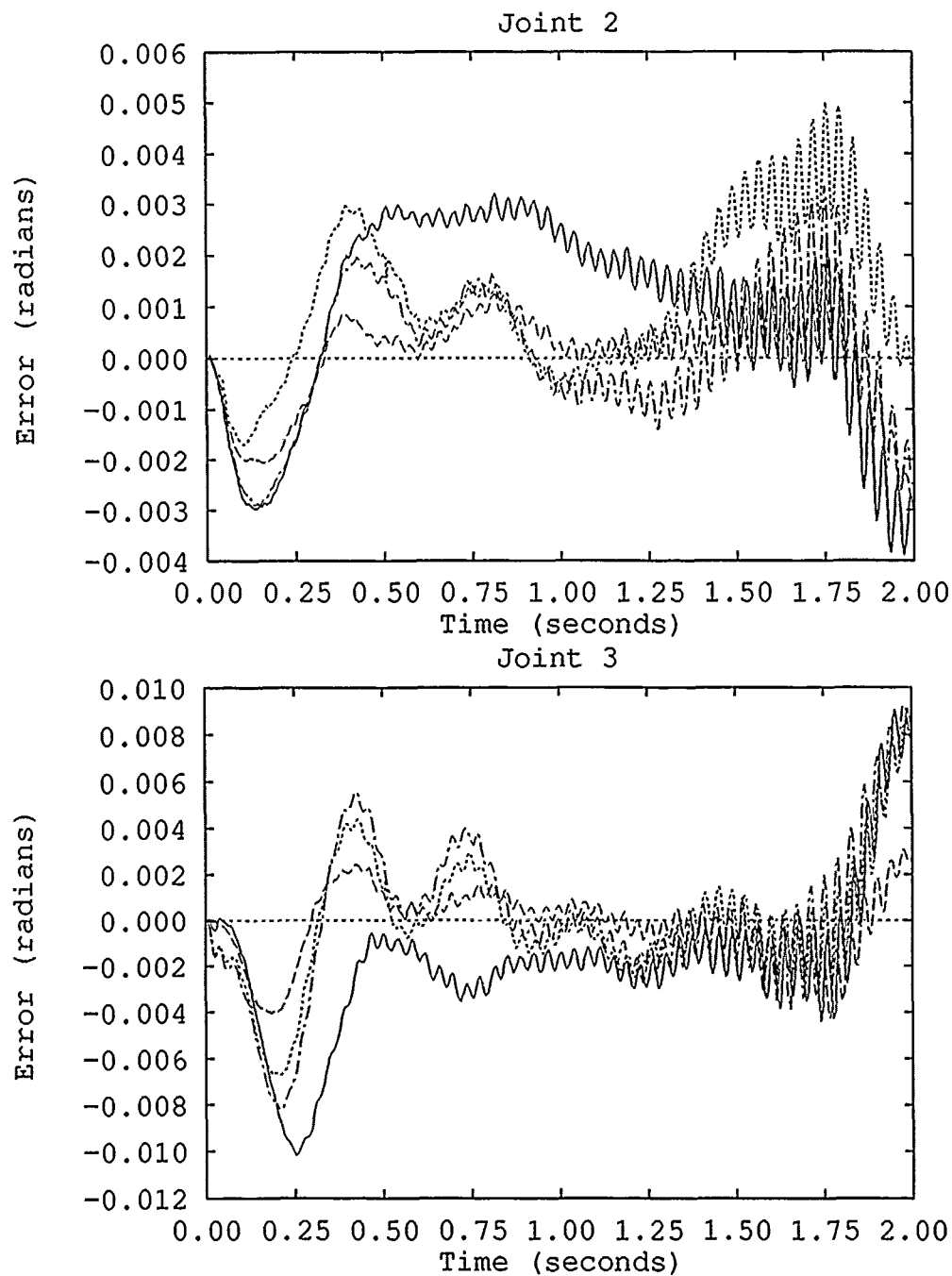


Figure 4.8. AMBC/H vs. AMBC/PBPA - Trajectory 3 with 2KG Payload

—	AMBC/H - Run 1	AMBC/PBPA - Run 1
----	AMBC/H - Run 5	-.-.-.-	AMBC/PBPA - Run 5

Table 4.3. Comparison of \hat{a} Values; 16 Estimated Parameters

Parameter Number	Physical Values	After Traj 1, Run 5	After Traj 2, Run 5	After Traj 3 Run 5
1	28.477	3.5538	3.641	1.9504
2	-9.66729	-51.5196	-50.0590	-49.4823
3	7.52559	-6.7353	-6.2001	-6.8044
4	6.820	-0.1414	0.0082	0.4095
5	5.950	2.1692	4.3580	4.0610
6	4.5	-6.4271	-5.8339	-5.7234
7	3.91	6.9872	6.1118	6.3920
8	3.5	-0.4327	-0.2136	0.1080
9	3.5	1.6702	-0.5122	-0.6259
10	1.49726	-0.4741	0.4858	0.5110
11	1.32209	6.8264	7.1849	7.0275
12	-1.29028	4.4262	4.4432	4.3942
13	-1.09687	5.1568	5.6519	5.0783
14	0.680048	5.4671	5.4393	5.5478
15	-0.618251	5.5158	5.4455	5.4122
16	-0.460108	-4.6204	-4.5763	-4.5672

ordered by sensitivity, if it is desired to reduce the number of estimated parameters, one can simply remove the appropriate number of parameters from the end of p^* . When the number of estimated parameters decrease, the number of non-estimated parameters increase correspondingly (refer to section 4.2.2). Figures 4.9-4.11 illustrate the effect of reducing the number of estimated parameters. It can be seen that minimal degradation in tracking accuracy occurs until only seven parameters are estimated. Table 4.4 sets out the final \hat{a} values for the same trajectories as were used in Table 4.3.

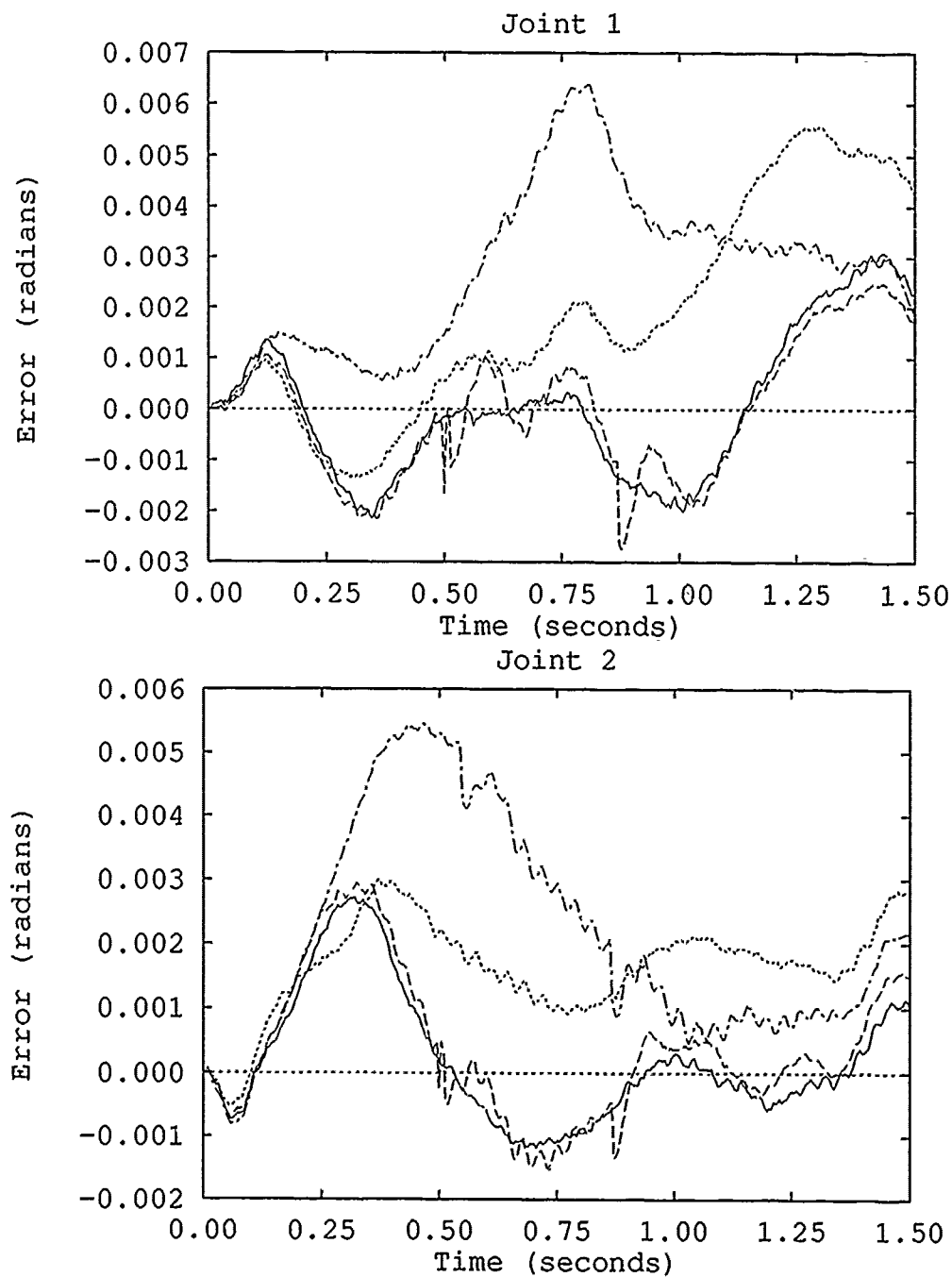


Figure 4.9. Effect of Parameter Reduction: Fifth Run, Trajectory 1

—	16 Parameters	10 Parameters
- - -	13 Parameters	- · - · -	7 Parameters

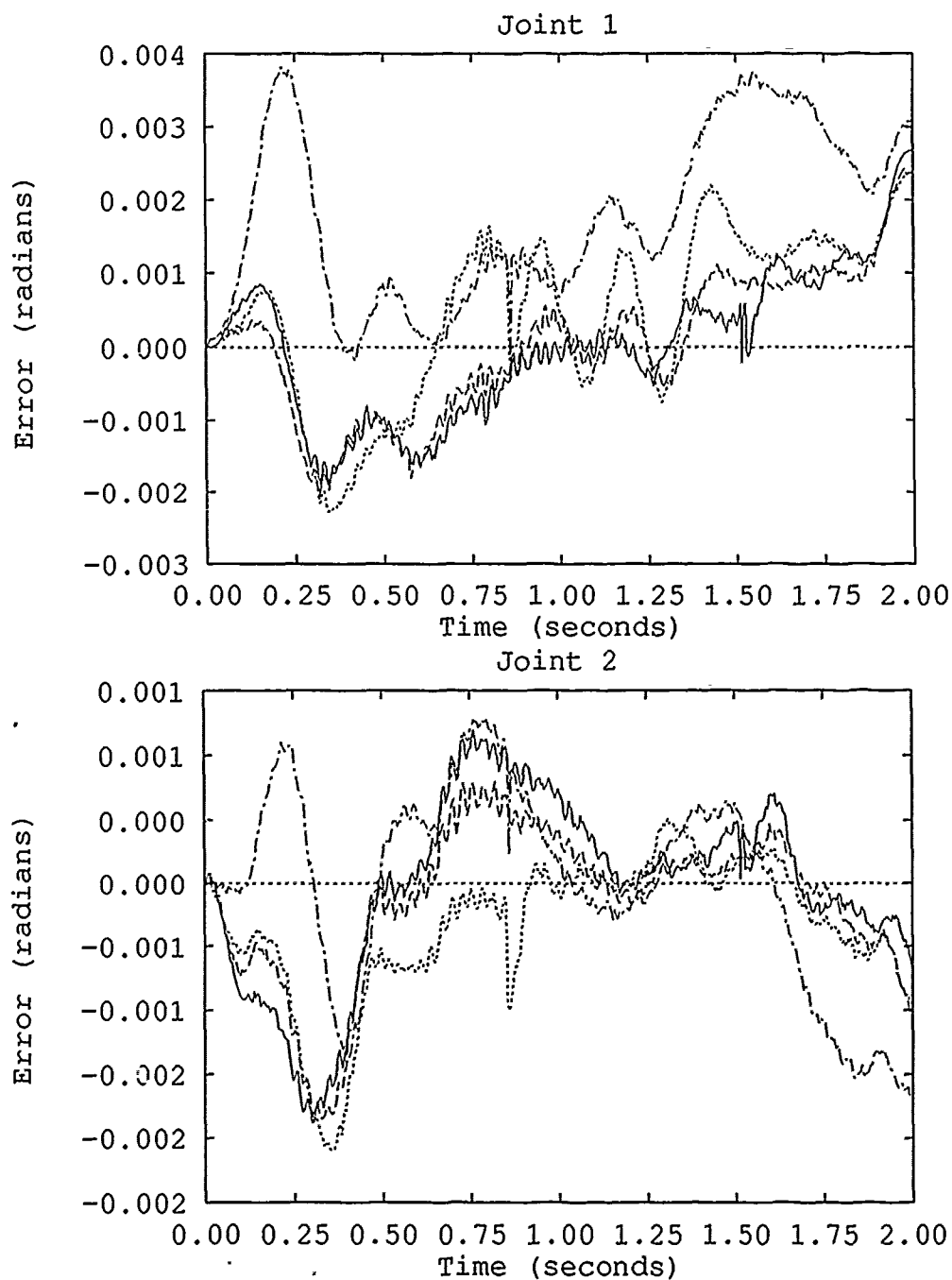


Figure 4.10. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2

—	16 Parameters	10 Parameters
- - -	13 Parameters	- . - . -	7 Parameters

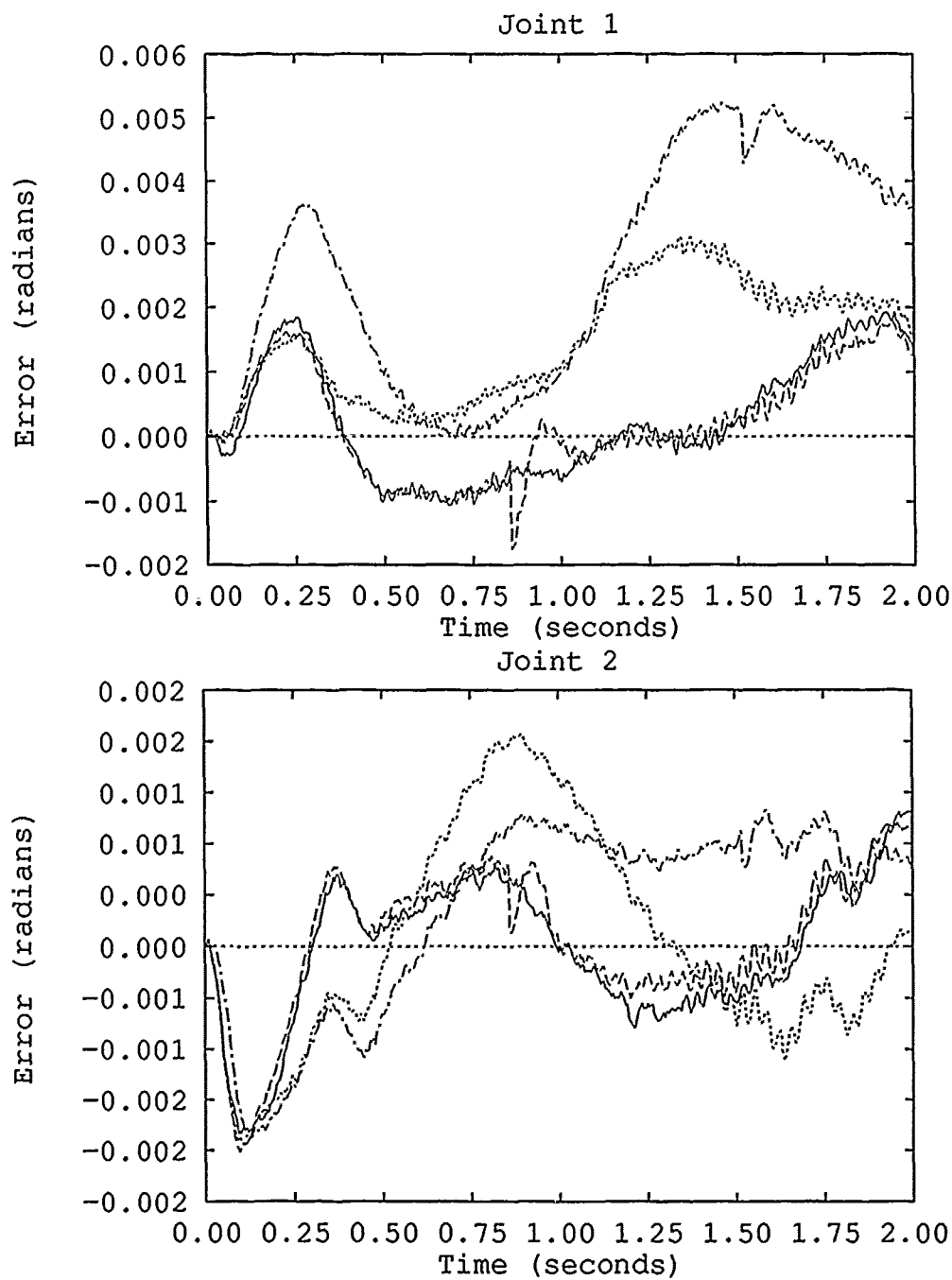


Figure 4.11. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3

—	16 Parameters	10 Parameters
- - -	13 Parameters	- - - -	7 Parameters

Table 4.4. Comparison of \hat{a} Values; 7 Estimated Parameters

Parameter Number	Physical Values	After Traj 1, Run 5	After Traj 2, Run 5	After Traj 3 Run 5
1	28.477	0.2242	3.641	-1.3817
2	-9.66729	-55.5330	-50.0590	-52.4891
3	7.52559	-5.4105	-6.2001	-6.3326
4	6.820	-0.8487	0.0082	1.0310
5	5.950	2.0968	4.3580	4.6289
6	4.5	-5.0666	-5.8339	-3.3298
7	3.91	8.2137	6.1118	5.6362

This simple method of parameter reduction is another advantage of AMBC/PBPA over AMBC/H. As previously stated, if it were desired to reduce the number of parameters estimated on line, one could simply use nominal physical values for the non-estimated elements of the AMBC/PBPA parameter vector. This simple reduction technique is impossible with AMBC/H - one cannot (without extensive knowledge of the manipulator dynamics) judge the relative importance of the parameters.

One surprising result came out of this parameter reduction exercise. In all cases, as expected, performance decreased as less parameters were estimated (adapted). However, in some cases, when the number of estimated parameters is decreased from seven to four, tracking accuracy actually increases (Figures 4.12 and 4.13). This result is incongruent with the expected results and bears further investigation.

4.3.5 Importance of Accurate Knowledge of Physical Values

Adaptive algorithms, as a whole, may enable a controller to overcome lack of a priori knowledge of the dynamics of a manipulator. This section is to address the importance of accurate knowledge of the physical values of the manipulator under study (e.g. masses, link lengths, inertias). For this test, the physical values, as given by Tarn and Bejczy are varied by $\pm 20\%$ [27]. P^* was then re-normalized and used in place of the original Γ^{-1} (refer to sections 3.2.2 and 4.2.3). Figures 4.14 and 4.15

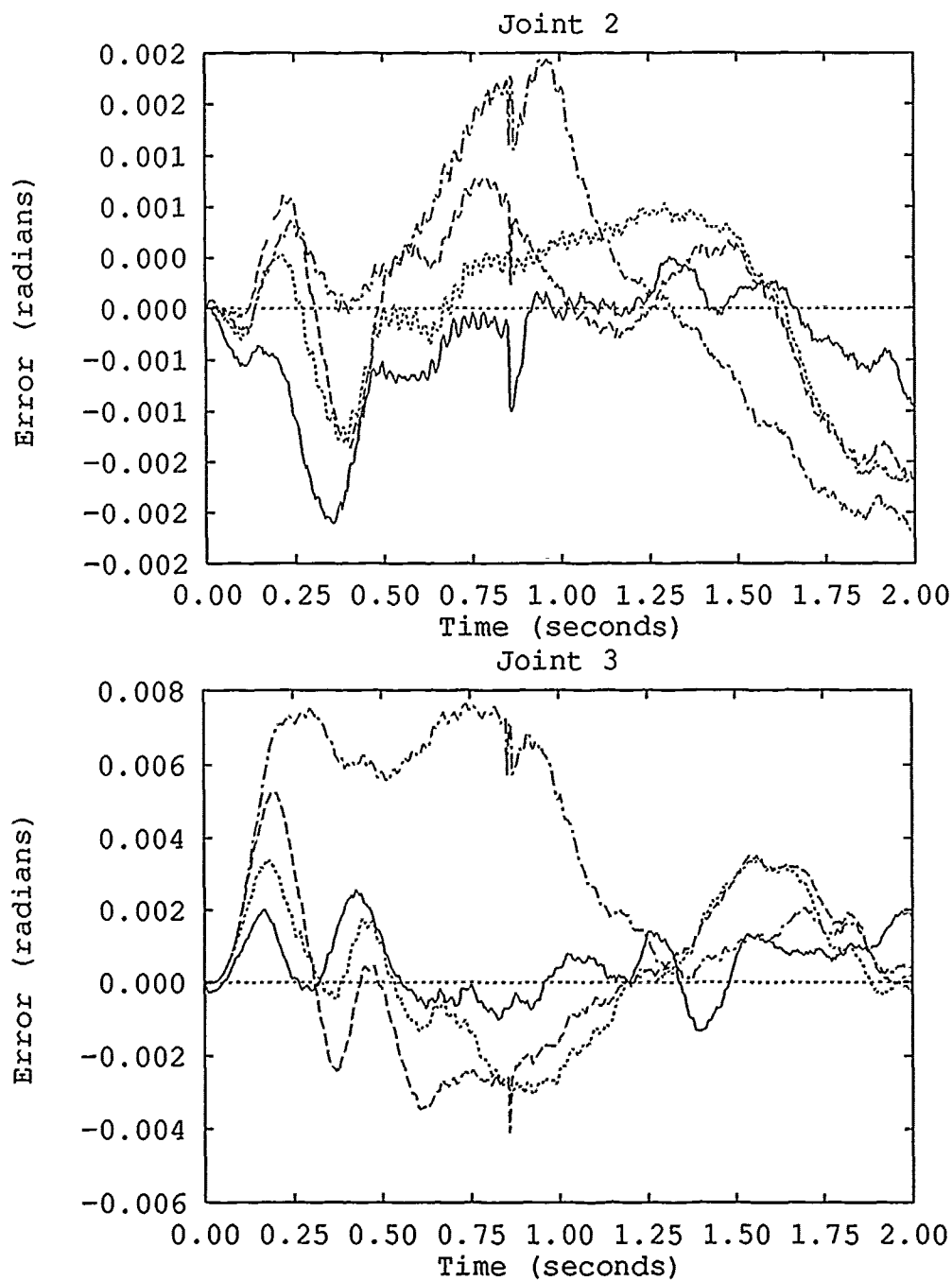


Figure 4.12. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2

—	10 Parameters	4 Parameters
---	7 Parameters	-.-.-.-	2 Parameters

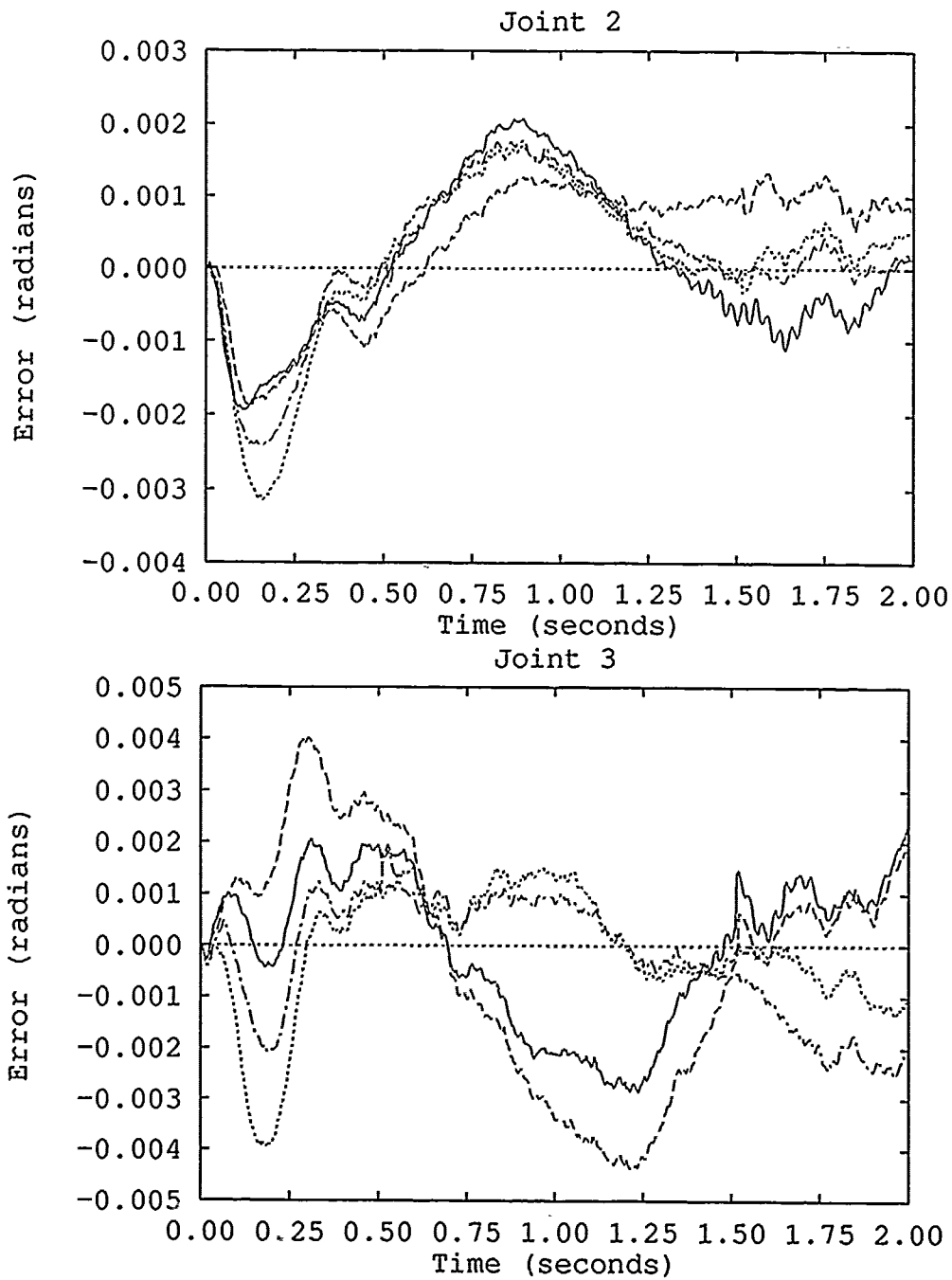


Figure 4.13. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3

—	10 Parameters	4 Parameters
- - -	7 Parameters	- · - · -	2 Parameters

show that the AMBC/ PBPA controller is somewhat sensitive to accurate knowledge of physical system values; however, this a priori knowledge is not absolutely essential to operation. Figure 4.14 demonstrates an oscillation on the -20% error plot. This oscillation can be squashed by lowering the scale factor slightly, with a corresponding increase in time to steady state error.

4.3.6 Adaptation versus Pattern Learning

The previous sections have demonstrated that the AMBC/PBPA algorithm can adapt successfully to new trajectories. Another question to be asked is whether the algorithm is adapting during the run, even after steady state error is reached. Figure 4.16 addresses this issue. For this test, the indicated trajectory was run until steady state error was reached. After steady state error occurred, the adaptation was turned off ($\Gamma^{-1} = 0$) causing the trajectory to be run with a constant feedforward torque applied. Figure 4.16 shows that adaptation is occurring during the trajectory, even after steady state error is reached. This is an indication that the algorithm has not simply 'learned' a pattern.

4.3.7 Soft PD Gains

All of the testing up to this point has used stiff PD feedback gains (ref 4.1) in conjunction with the AMBC algorithm. This portion of the test plan adjusted the PD gains to the 'soft' values shown in Table 4.5. The use of high feedback gain initially improves PUMA tracking performance (e.g. before adaptation) [18, 15]. Figure 4.17 shows how this change affects performance. Initially, tracking error is higher, but steady state error is essentially the same (as compared to the high PD gain plots).

If these results are compared to those shown in Figure 2.1 (Trajectory 1, stiff PD gains), two items should be noted. First, the relative tracking performance of the two algorithms remains the same - AMBC/II is better by an approximate factor

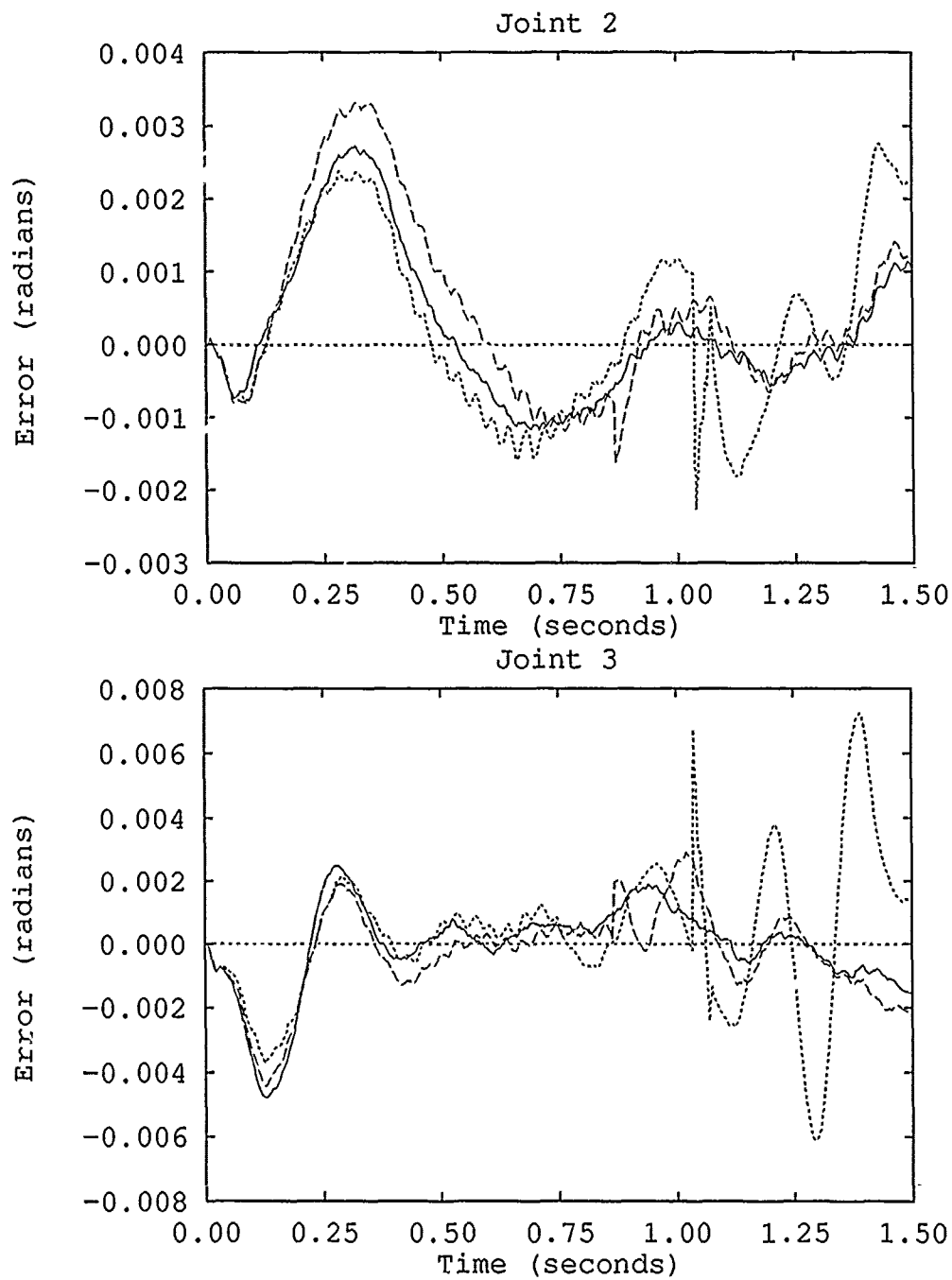


Figure 4.14. Effect of Varying Physical Parameter Values: Trajectory 1, Run 5

—	Best Estimate	----	Best Estimate +20%
.....	Best Estimate -20%		

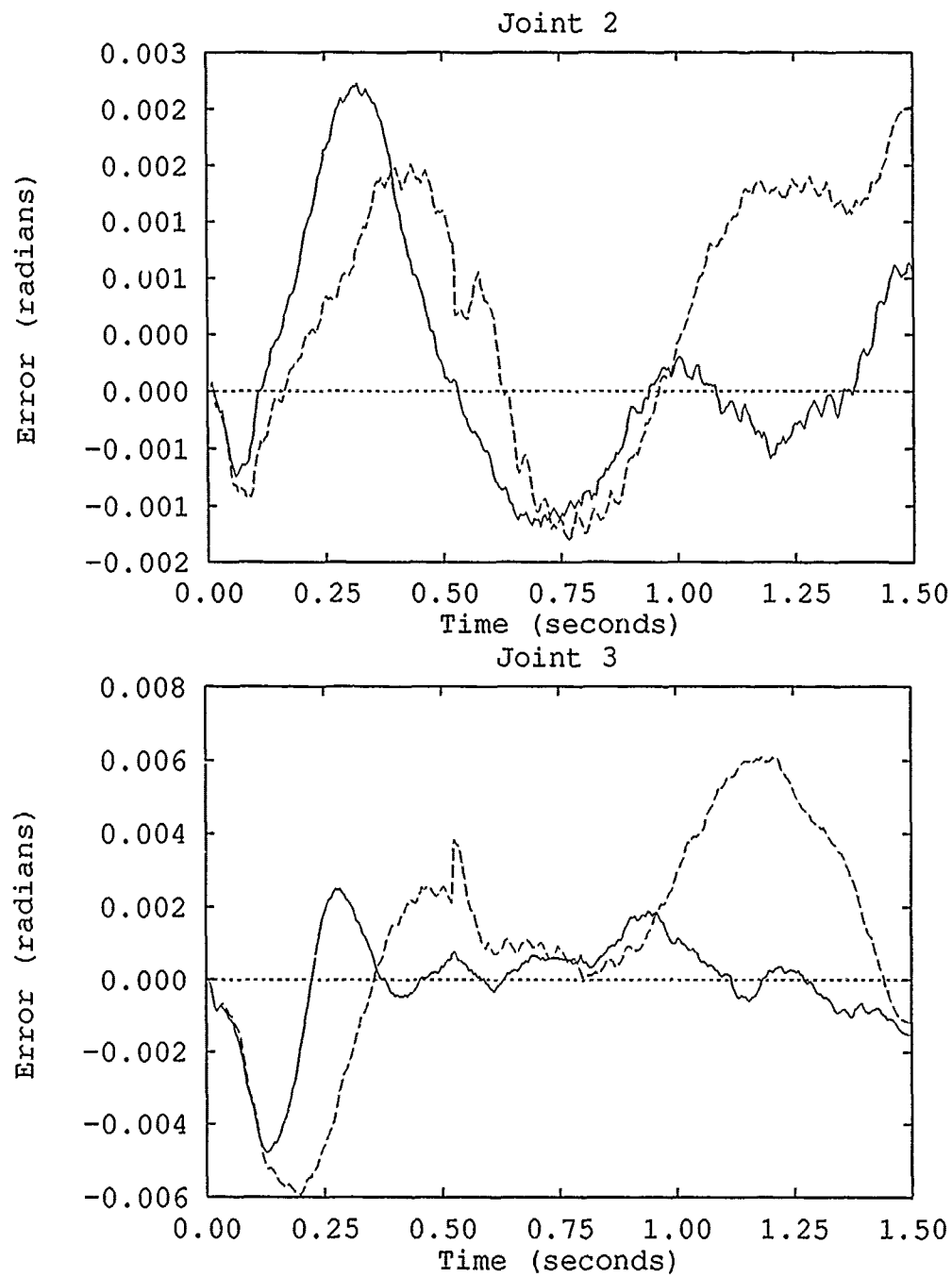





Figure 4.15. Effect of Varying Physical Parameter Values: Trajectory 3, Run 5

	Best Estimate		Best Estimate +20%
	Best Estimate -20%		

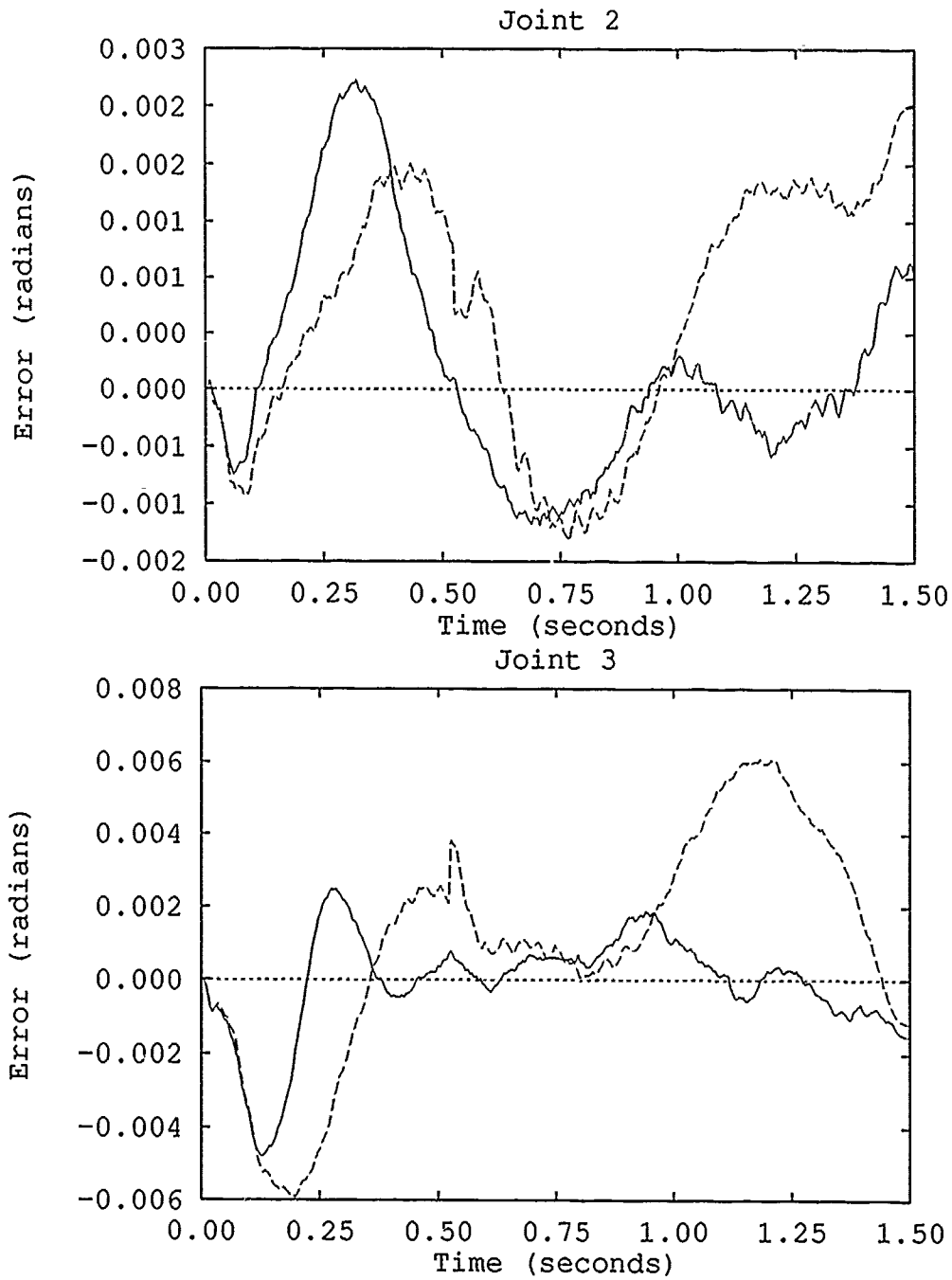


Figure 4.16. Effect of Discontinuing Adaptation - Trajectory 1

—	AMBC/PBPA with Adaptation
- - -	AMBC/PBPA without Adaptation

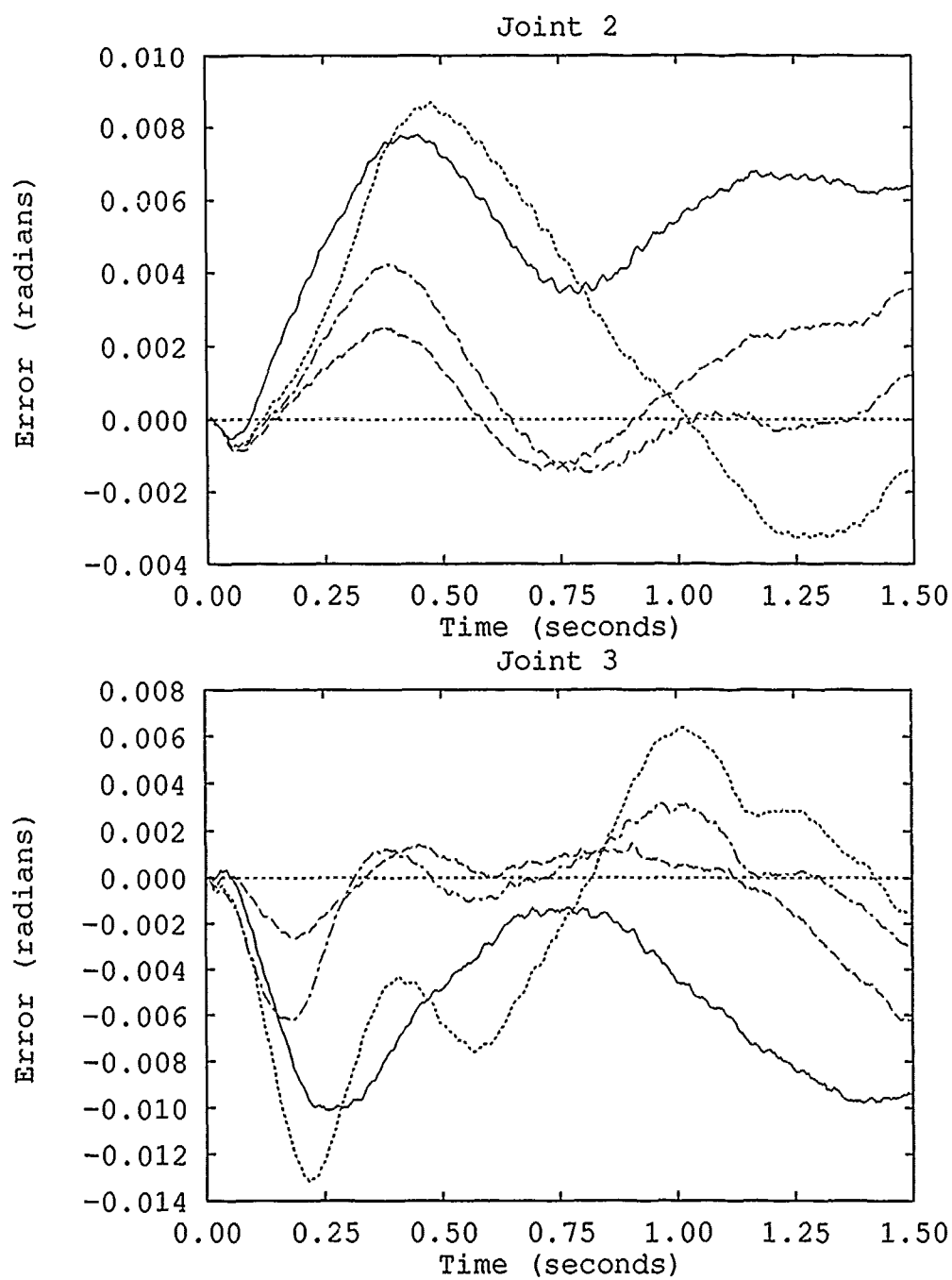


Figure 4.17. Effect of Low PD Gains, Trajectory 1, No Payload

—	AMBC/H, Run 1
- - - -	AMBC/H, Run 5
.....	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 5

Table 4.5. Soft PD Feedback Gains

	Joint 1	Joint 2	Joint 3
K_p	250.0	520.0	96.0
K_v	72.0	130.0	25.0

of two. Secondly, while the initial error is much higher for the soft PD gain scenario, steady state error for both cases (soft and stiff PD gains) is close to the same profile.

4.3.8 Two Way Tracking Evaluation

The testing described in the previous sections moved the manipulator through unidirectional trajectories. Figure 4.18 shows the effect of a bidirectional trajectory. This particular test moved the arm through Trajectory 1, then back to the initial position after a midpoint pause of 500 msec. This figure shows that the AMBC/PBPA algorithm still provides excellent tracking accuracy. The spike just past midpoint is due to drive train stiction, as the manipulator changed direction.

4.3.9 Very Slow Tracking

This scenario (Figure 4.19) is included to provide a link to previous AFIT research [15]. As with previous scenarios, AMBC/PBPA provides essentially the same tracking accuracy as does the AMBC/H algorithm. This particular scenario is valuable in showing that even with the stiff PD gains, the manipulator can successfully track without significant vibration.

4.3.10 Effect of Varying the Γ^{-1} Scaling Factor

At several points during this report, the scaling factor for the AMBC/PBPA algorithm has been discussed. Figures 4.20 and 4.21 illustrate the importance of this scaling factor. As stated in section 4.2.3, the scaling factor used for the previous plots was 0.02. Again, this was the largest value that could be used while allowing tracking

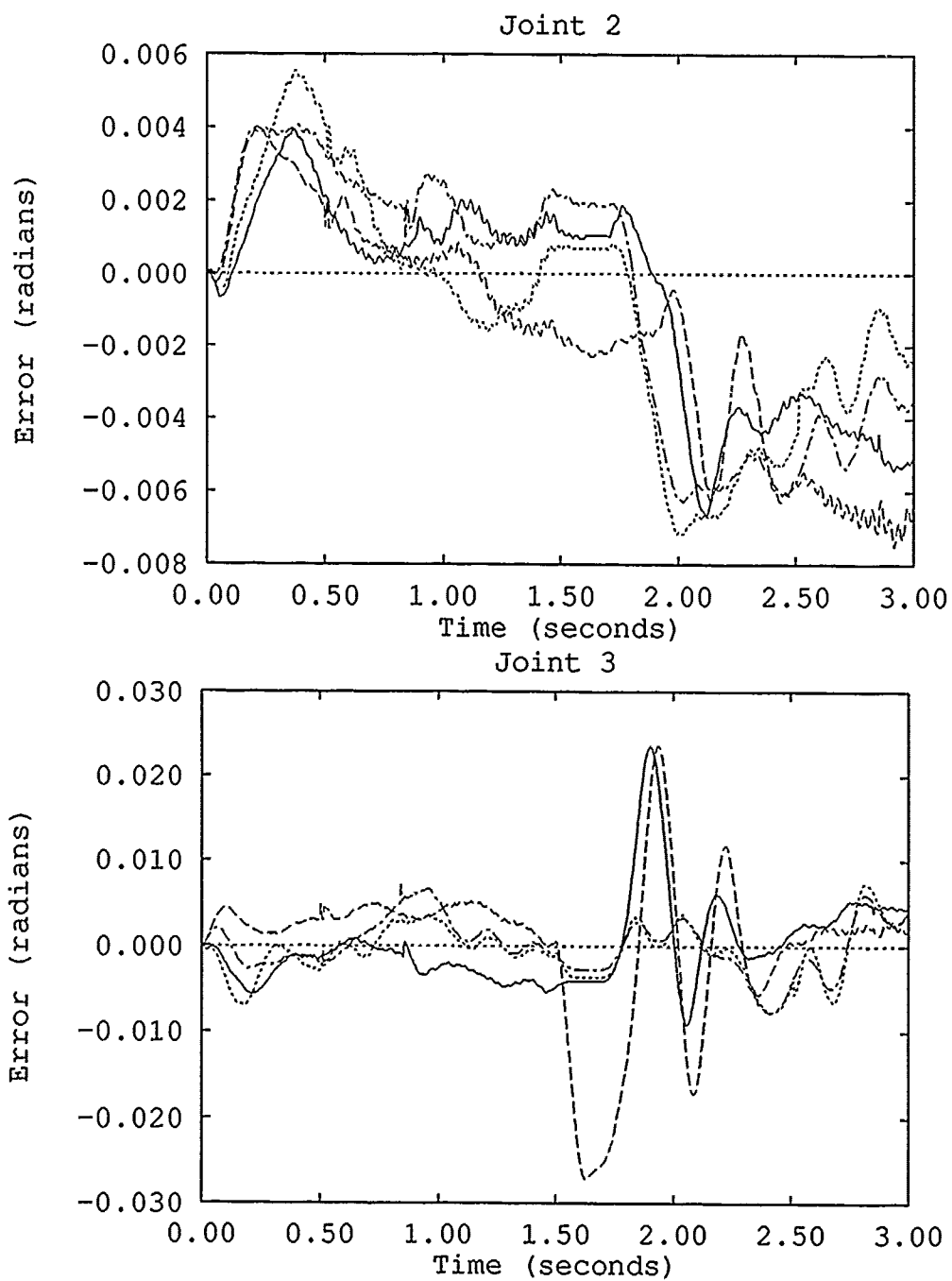


Figure 4.18. Two Way Trajectory Tracking Evaluation, No Payload, Trajectory 1

—	AMBC/H, Run 1
----	AMBC/H, Run 3
.....	AMBC/PBPA, Run 1
-.-.-.	AMBC/PBPA, Run 3

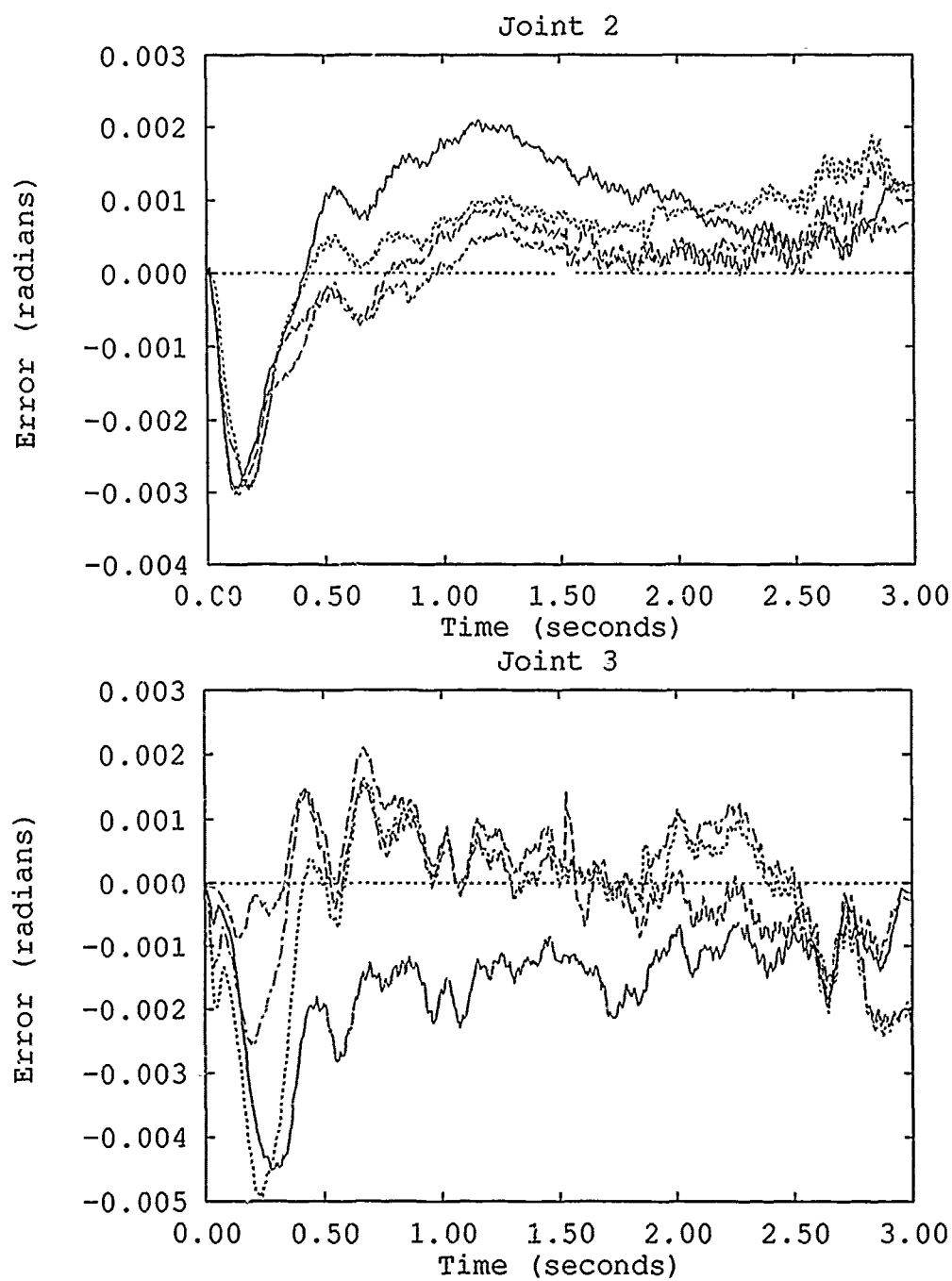


Figure 4.19. Very Slow Tracking Evaluation, No Payload, Trajectory 3

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
.....	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 5

over the entire trajectory suite. This value was determined experimentally; however, this tuning process took minutes as opposed to the months required to tune the AMBC/H Γ^{-1} . If the scaling factor is set too high, instability occurs. If the scaling factor is too low, time to steady state error increases. The scaling factor chosen for some arbitrary task is trajectory dependant. That is, for a repetitive, single motion, the scaling factor can be set somewhat higher, causing quick convergence to steady state error. An application that has many associated motions would require a smaller scaling factor - this trades off time to steady state error against flexibility. Regardless, changing the scale factor is only the matter of a few keystrokes.

4.3.11 Numerical Comparison

In a recent paper by Whitcomb, Rizzi and Koditschek [19], they state that a visual comparison of graphs, such as those used in this study, often become an act of aesthetic judgement rather than empirical analysis. They propose that using a scalar norm (ν^2) would be a preferable alternative since it would provide a single, numerical measure of tracking performance for the entire error plot [19]. Their equation is

$$\nu^2[e(t)] = \left(\frac{1}{t} \int_{t_0}^t \|e(t)\|^2 dt \right)^{\frac{1}{2}} \quad (4.1)$$

where $e(t)$ is a selected scalar (or vector) valued tracking error. The norm measures the root-mean-square 'average' of the tracking error, thus a smaller ν^2 represents a smaller tracking error [19]. This equation was applied against all of the trajectories and selected runs and the results are compiled in Table 4.6. For each of the cases shown, the equation was applied to the error profile for the equivalent AMBC/H and AMBC/PBPA run. The resultant number for AMBC/H was then divided by the resultant number for AMBC/PBPA. The final, single scalar value is the ratio of the AMBC/H error to the AMBC/PBPA error. Therefore, if $\nu^2 = 1$ it indicates that both algorithms delivered the same error performance, $\nu^2 < 1$ indicates that AMBC/H performed better and if $\nu^2 > 1$, AMBC/PBPA performed better. One

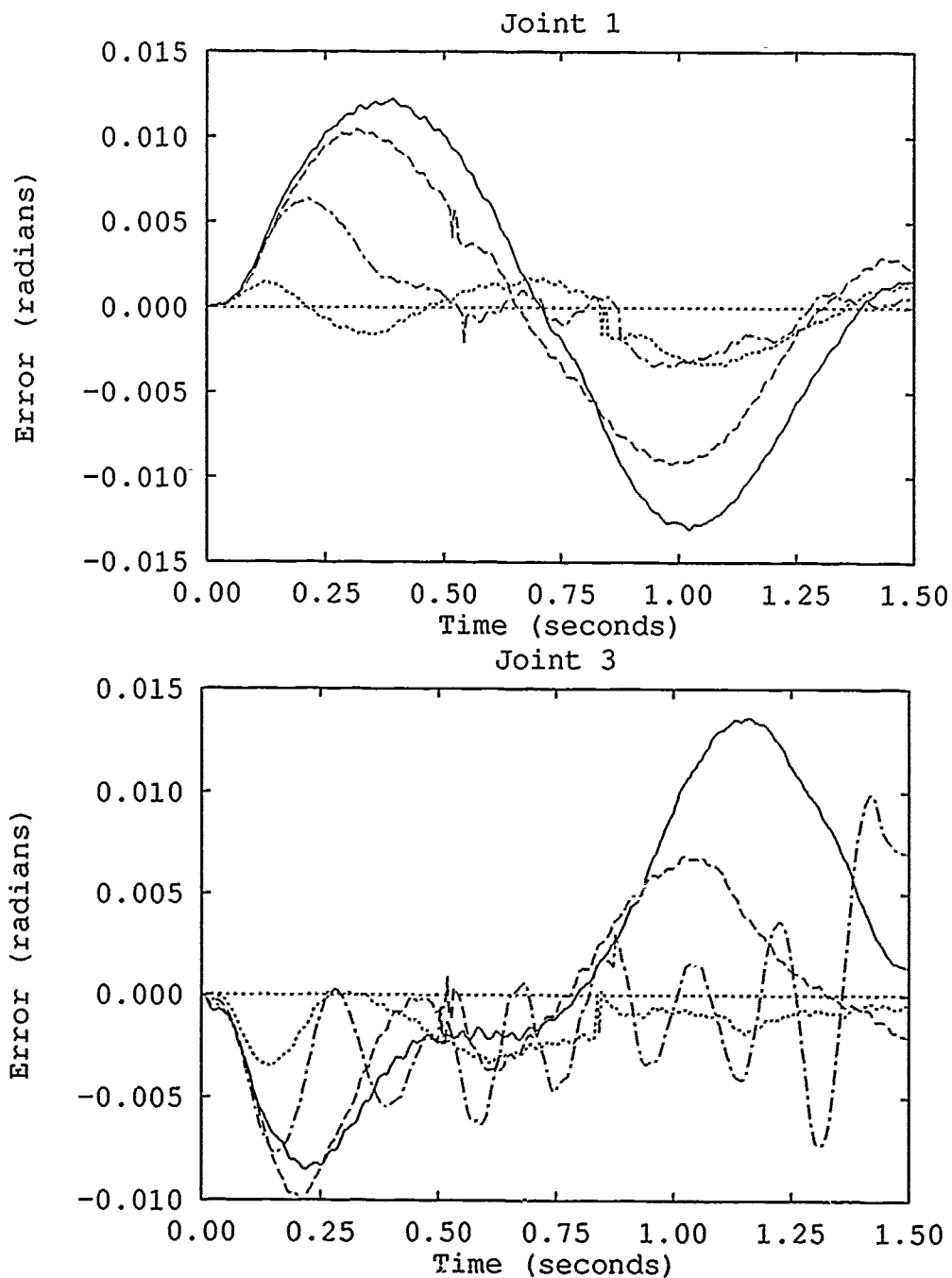


Figure 4.20. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 1

—	Scale Factor = 0.001
- - -	Scale Factor = 0.005
...	Scale Factor = 0.01
- . - .	Scale Factor = 0.03

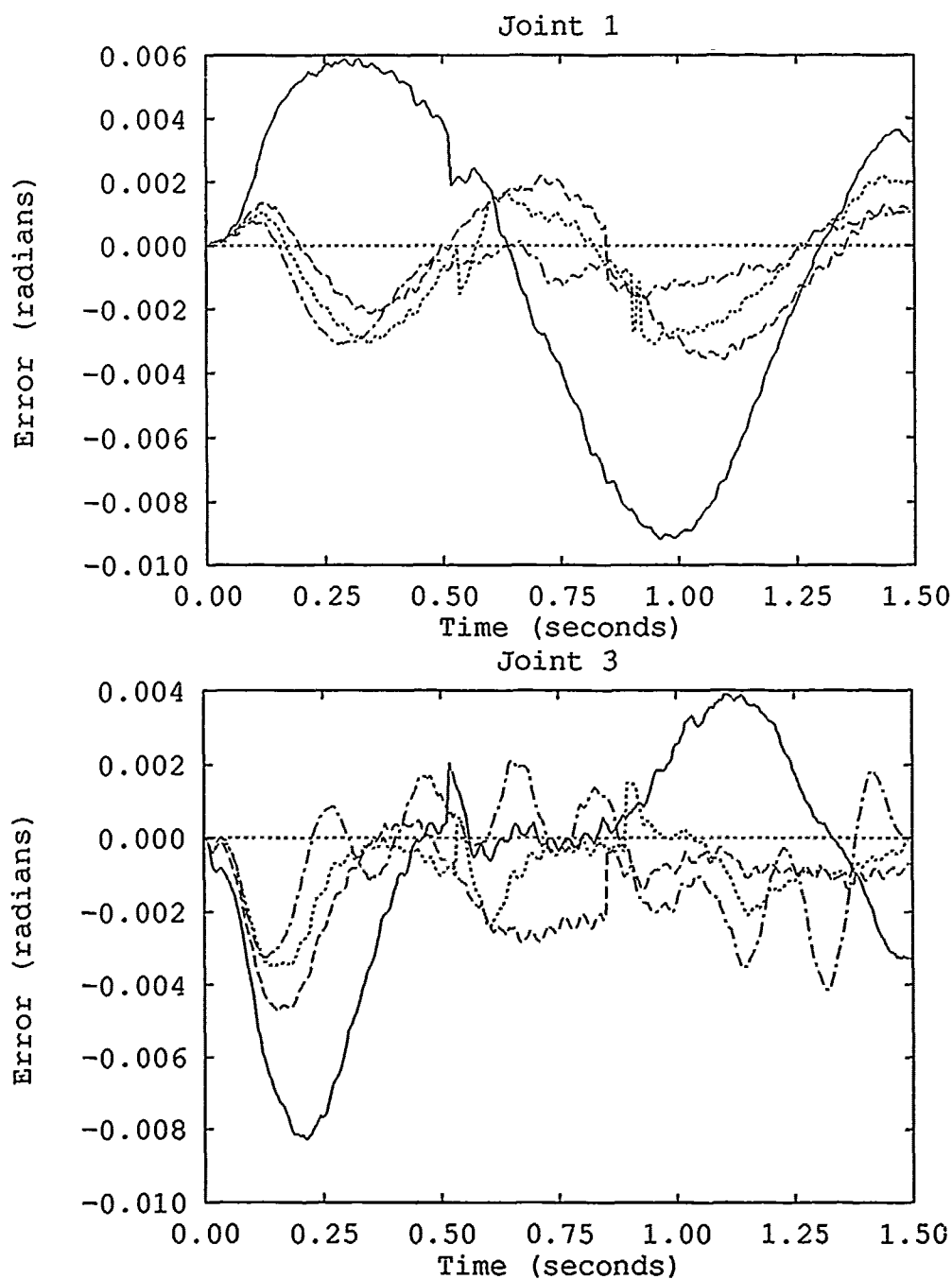


Figure 4.21. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 5

—	Scale Factor = 0.001
- - -	Scale Factor = 0.005
.....	Scale Factor = 0.01
- . - .	Scale Factor = 0.03

piece of aberrant behavior should be noted - AMBC/H was unable to track Trajectory 4 after 3 runs; therefore, the values shown for that particular case are unusually large.

Table 4.6. Comparison of Tracking Errors using Equation 4.5

Trajectory	Run # 1			Run # 5		
	Joint 1	Joint 2	Joint 3	Joint 1	Joint 2	Joint 3
1	0.3248	0.9370	1.1189	0.6535	0.5302	0.5984
2	0.2528	0.1650	0.2883	0.6198	0.7052	0.7735
3	0.9434	1.3639	1.2248	0.7177	1.3013	0.6257
4	0.8876	0.7236	0.5491	26.2056	25.2412	19.0522
5	0.8655	0.3345	1.6882	1.5449	0.6516	1.2481
6	0.6510	0.6110	0.4085	0.9505	0.4219	0.6420
1 (2Kg)	0.7072	1.7161	1.1891	0.6232	0.6516	0.4915
3 (2Kg)	0.4087	1.1055	1.3721	0.6630	0.7152	0.4699

Table 4.6 reinforces what can be seen in all of the figures that compare AMBC/H to AMBC/PBPA - that is, while AMBC/H usually provides somewhat better error tracking, both algorithms perform to the same order of magnitude. Except for the aberrant case of Trajectory four, at no time does either algorithm outperform the other by more than a factor of four - typically, AMBC/H outperforms AMBC/PBPA by a factor of two. This difference in performance is a significant amount (e.g. two times as good); however, it must be remembered that these errors are in thousandths of radians. At such a level of accuracy, a difference of two to one would be insignificant for most industrial applications. Additionally, it must be remembered that the AMBC/PBPA algorithm was tuned in minutes versus months for the AMBC/H algorithm. In many applications, this time savings may be an acceptable trade-off for the decrease in performance.

4.4 The Utah/MIT Dexterous Hand

As stated earlier, the intention was to cross check this procedure on a second manipulator - specifically, the UTAH/MIT Dexterous Hand (UMDH). Since the time of the study performed by Rainey, [22] the computer interface to the UMDH (along with the associated operating system) has changed. This hardware change necessitated software changes. Unfortunately, due to delays in the hardware becoming operational, the tie does not remain to complete this part of the study. To date, PBPA has been completed on the UMDH, using the same 2 DOF configuration as Rainey [22]. In addition, the results of the PBPA have been used to create an AMBC algorithm for the hand. Only the test and evaluation of this algorithm remains to be done. Completion of this project should be considered as an area for future study.

4.5 Summary

A exhaustive comparison of the two algorithm versions has been performed. These results indicate that actually able to provide the minimal set of parameters needed by and Adaptive Model-Base Control algorithm. Furthermore, this method of determining the minimal parameters also allows for a quick and simple method of tuning the AMBC controller for optimal performance.

V. Conclusions and Recommendations

5.1 Conclusions

Principal Base Parameter Analysis provides a straightforward, mathematical method for determining the base parameter set of a vertically articulated manipulator. More importantly, this analysis allows the user to easily fine tune the feed-forward gain for optimal performance in a logical, methodical manner. This Tuning/Parameter Reduction technique has two advantages. First, the straightforward procedure frees the AMBC designer from needing extensive knowledge of the manipulator dynamics. The second advantage is the great time savings during the tuning process. Starting with just the torque equations, a designer with knowledge of the PBPA process and a symbolic mathematics software package can design and implement an AMBC algorithm in just a few short hours. This quick design may not meet or exceed the performance of an AMBC algorithm tuned over the course of months, but for many industrial applications, this performance degradation would be acceptable in light of the time savings.

The AMBC/PBPA closely paralleled the performance of the AMBC/H algorithm in every test scenario presented. In the areas of simple trajectory tracking, robustness to payload variation, very slow tracking et. al., the AMBC/PBPA algorithm performed adequately. Furthermore, the AMBC/PBPA converged to steady state error in three to five runs, as did the AMBC/H algorithm. These results lead to the conclusion that PBPA is a viable option to be used in the design and tuning of an Adaptive Model-Based Controller - with the caveat that the simplicity will be at the cost of some performance degradation.

5.2 Recommendations for Future Study

Often, advanced study efforts raise more questions than they answer. This study was no different. While PBPA may provide the push needed to move AMBC

out of the lab and onto the factory floor, not every question has been answered. This section will identify several areas which could be explored in more depth.

5.3 Disparity Between AMBC/H and AMBC/PBPA

As was discussed in Chapter 4, AMBC/H typically outperforms AMBC/PBPA by a ratio of 2 to 1. Theoretically, since they are both derived from the same torque equations, they should both provide the same approximate performance. Since this is obviously not the case, where is the discrepancy? Two possible causes of this discrepancy are the physical values used and the neglected dynamics. First, no special effort was made to measure the physical parameters of the manipulator. This approach was deemed acceptable since an AMBC algorithm is to overcome this lack of accurate knowledge. Secondly, the only drive system information used was viscous and coulombic friction. Consequently, any neglected drive system information was accounted for in some other unrelated parameter.

5.4 AMBC/PBPA Fine Tuning

The attempt of the test and evaluation of the PBPA design process was kept strictly in the realm of using one simple scale factor. If it were desired to improve the performance of the AMBC/PBPA algorithm, it would be a simple matter to heuristically fine tune the AMBC/PBPA algorithm further. While this type of fine tuning certainly does not fit in with the non-heuristic procedure developed in this study, it may be acceptable to expend some effort on fine tuning to improve performance. The suggested area for future study is to quantify the performance gain if a manual fine tune of the AMBC/PBPA algorithm is performed. A second, related area is to duplicate the test suite used in the study, using the maximum possible scale factor in each individual case. Many robots used in standard industrial application have a limited repertoire of motions. In such a case it would be acceptable to have a separate, optimal scale factor for each movement.

5.5 Cross Check of Procedure on a Second Platform

As discussed earlier, the intent was to cross check the results of this study on a second robotic platform. Due to delays in hardware development, this sideline was not feasible. This particular area of study is of prime importance to prove that the PBPA design technique is not platform specific. The second robot that was designated to be the cross check platform was the Utah/MIT Dexterous Hand (UMDH). This robot would be an excellent choice for this type of study since it's internal dynamics are not well known (as are the PUMA's). Consequently, any heuristic tuning done on the UMDH would be simple trial and error and not assisted by any intuitive jumps of logic. If the PBPA developed AMBC algorithm could out-perform a heuristically tuned AMBC algorithm, it would be substantial proof that PBPA is not just a quixotic idea.

The pursuit of these recommendations could advance PBPA as a possible AMBC design approach into an integral part of every robotic design toolbox.

5.6 Summary

The PUMA 560 has been the platform used for extensive study of Adaptive Model-Based Control algorithms at the Air Force Institute of Technology [16]. During the course of these studies, researchers have been able to heuristically tune the adaptation gains for optimal performance. Furthermore, they have been able to reduce the parameter set used with no negative effect on the tracking accuracy. The purpose of this study was to apply the Principal Base Parameter Analysis technique to the PUMA 560, incorporate the reduced parameter set into the existing AMBC algorithm, then contrast the tracking accuracy achieved with the PBPA reduced parameter set against previous results. The final result is that there is a method, essentially free of heuristics, for developing and tuning an Adaptive Model-Based Controller. The simple, straightforward procedure detailed in this report may be

the key that allows AMBC to move out of the laboratory and into operational applications.

Bibliography

1. C.G. Atkeson C.H. An and J.M. Hollerbach. Estimation of Inertial Parameters of Rigid Body Links of Manipulators. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 84-89, 1985.
2. D.E.Schmitz D.B.Stewart and P.K.Khosla. CHIMERA II Real-Time Programming Environment. Technical report, Carnegie Mellon University, June 1991. Dept. of Elect. and Comp. Eng.
3. M. Gautier. Numerical calculation of the base inertial parameters of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1020 - 1025, 1990.
4. M. Gautier and W. Khalil. Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots. *IEEE Transactions on Robotics and Automation*, 6(3):368-373, 1990.
5. M. Ghodoussi and Y. Nakamura. Principal Base Parameters of Open and Closed Kinematic Chains. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 84-89, 1991.
6. K. Osuka H. Mayeda and A. Kangawa. A new identification method for serial manipulator arms. In *The Ninth IFAC World Congress*, pages 2429-2434, 1984.
7. G. Warshaw H. Schwartz and T. Janabi. Issues in robot adaptive control. In *Proc. of ACC*, pages 2797-2805, 1990. May.
8. Koji Yoshida Hirokazu Mayeda and Koichi Osuka. Base Parameters of Manipulator Dynamic Models. *IEEE Transactions on Robotics and Automation*, 6(3):312 - 321, 1990.
9. IRONICS Inc. IV-3201 VMEbus Multiprocessing Engine: User's Manual, 1987.
10. SARCOS Inc. Hand electronics document package, March 1987.
11. S.C. Jacobsen, E.K. Iverson, D.F. Knutti, and K. B. Biggers. Design of the Utah/MIT Dexterous Hand. In *Proc. IEEE Conf. on Robotics and Automation*, pages 1520-1531, 1986. Vol. 3.
12. P. Khosla and T. Kanade. Parameters Identification of Robot Dynamics. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 84-89, 1985.
13. R.C.Gonzales K.S.Fu and C.S.G.Lee. *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill Book Company, 1987.
14. M. B. Leahy Jr. and P. V. Whalen. Adaptive Model-Based Control: An Experimental Case Study. ARSL-90-7, Air Force Institute of Technology, October 1990.

15. M. B. Leahy, Jr. and P. V. Whalen. Enhancements to Robotic Manipulator Trajectory Tracking. Technical Report ARSL-90-6, Submitted to the Journal of Robotic Systems, Air Force Inst. of Tech., June 1990. Dept. of Elect. and Comp. Eng.
16. M. B. Leahy Jr and P. V. Whalen. Direct adaptive control for industrial manipulators. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1666-1672, 1991.
17. M.B. Leahy Jr. Experimental analysis of robot control: A Performance Standard for the PUMA-560. In *Proc of IEEE 4th Symposium on Intelligent Control*, pages 257-264, September 1989.
18. M.B. Leahy Jr. Model-Based Control of Industrial Manipulators: An Experimental Analysis. *Journal of Robotic Systems*, 7(5):741-758, October 1990.
19. Alfred A. Rizzi Louis L. Whitcomb and Daniel E. Koditschek. Comparative Experiments with a New Adaptive Controller for Robot Arms. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2 - 7, 1991.
20. Yoshihiko Nakamura. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Publishing Company, 1991.
21. Romeo Ortega and Mark W. Spong. Adaptive Motion Control of Rigid Robots: A Tutorial. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 1575-1584, 1988.
22. Captain Lloyd W. Rainey. Digital Control of the UMDH: Initial Evaluation and Analysis. In *International Symposium on Intelligent Control*, page TBD, 1991.
23. Nader Sadegh and Roberto Horowitz. Stability and Robustness of a Class of Adaptive Controllers. *International Journal of Robotics Research*, 9(3):74-92, June 1990.
24. Captain Daniel J. Sims. Investigation of Adaptive Controllers for PUMA Trajectory Tracking. Master's thesis, Air Force Institute of Technology, Air University, 1991. GE/ENG/91J-50.
25. J-J. E. Slotine and W. Li. Adaptive Manipulator Control: A Case Study. In *Proc. of IEEE Int. Conf. on Robotics and Automation*, pages 1392-1400, 1987.
26. Mark W. Spong and M. Vidyasagar. *Robot Dynamics and Control*. John Wiley and Sons, 1989.
27. T.J. Tarn and A.K. Bejczy. Dynamic Equations for PUMA-560 Robot Arm. Technical Report SSM-RL-85-02, Washington University, St. Louis, MO, July 1985. Department of Systems Science and Mathematics.
28. Stephen Wolfram. *Mathematica: A System for Doing Mathematics by Computer*. Addison-Wesley Publishing Company, 1988.

Appendix A. *Comprehensive Set of Figures*

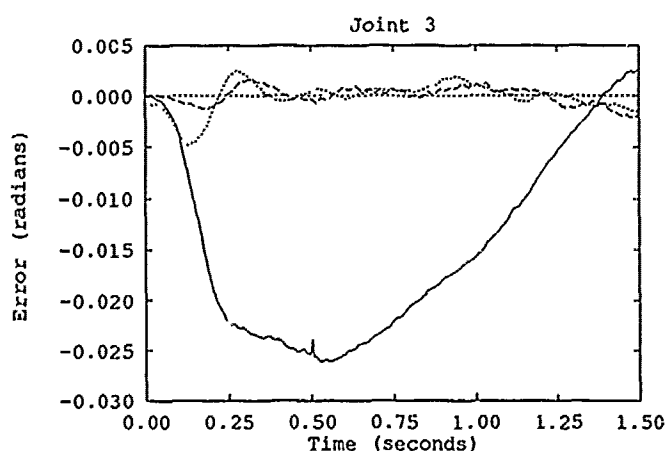
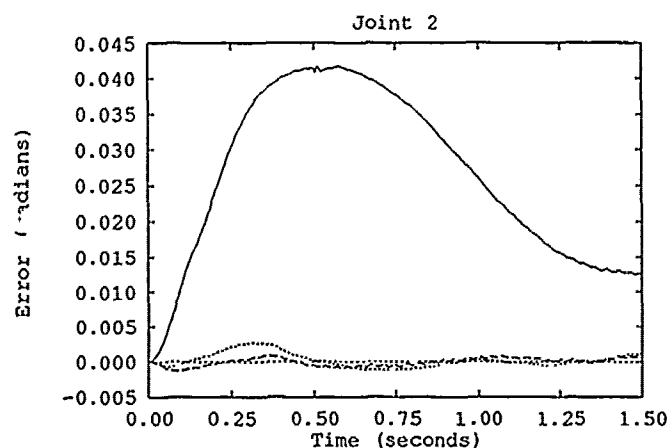
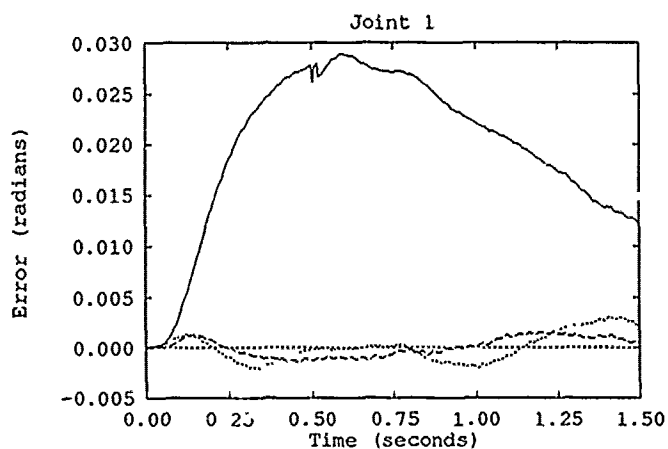


Figure A.1. PD Only vs PD with AMBC/H - Trajectory 1

—	PD
- - -	AMBC/H
.....	AMBC/PBPA

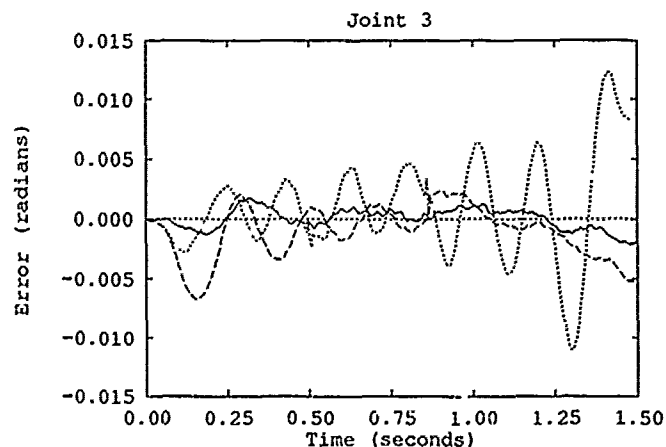
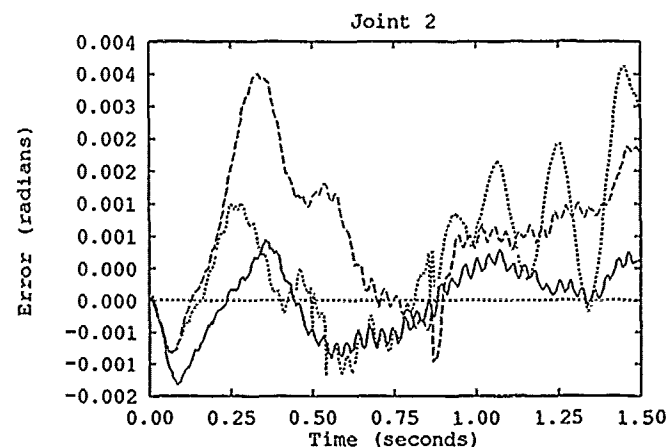
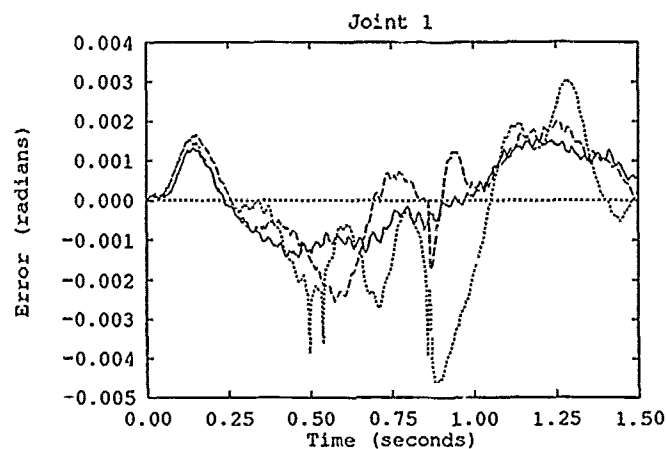


Figure A.2. Effect of setting AMBC/H Γ^{-1} to a Common Value: Fifth Run, Trajectory 1

—	Customized
- - -	$\Gamma^{-1} = 25$
.....	$\Gamma^{-1} = 37.5$

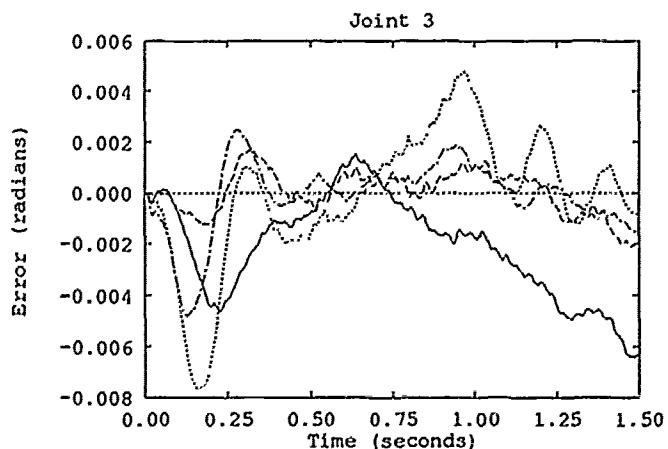
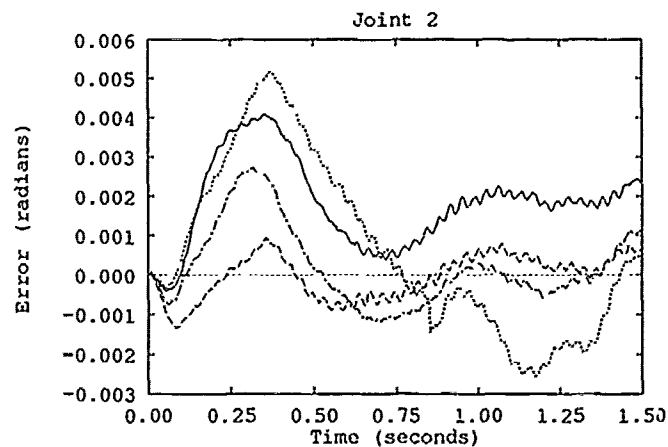
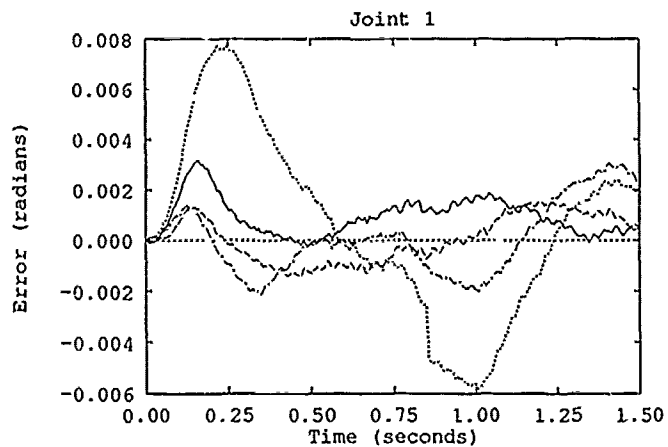


Figure A.3. AMBC/H vs. AMBC/PBPA- Trajectory 1

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
.....	AMBC/PBPA - Run 1
- · - ·	AMBC/PBPA - Run 5

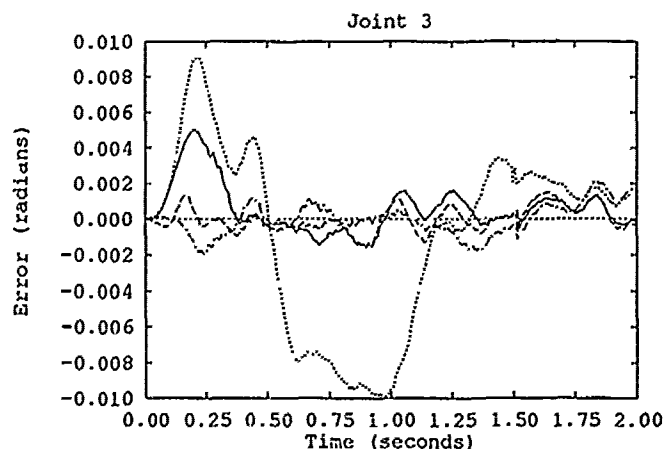
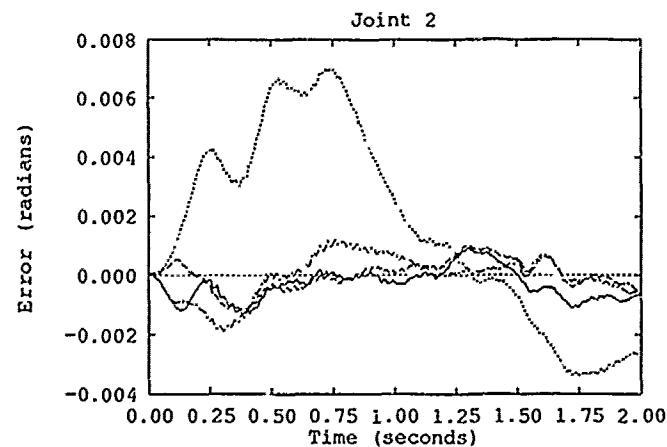
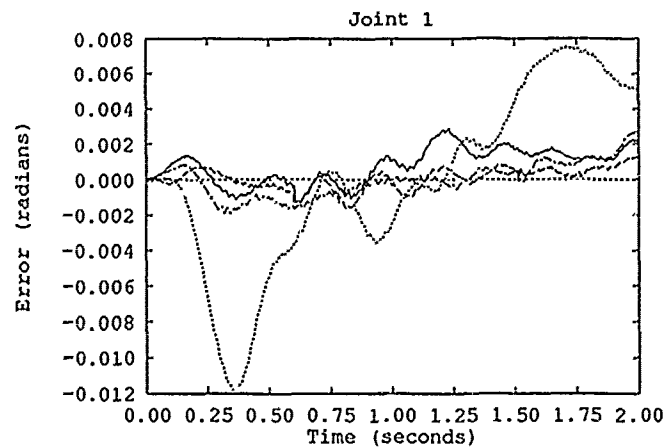


Figure A.4. AMBC/H vs. AMBC/PBPA - Trajectory 2

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
.....	AMBC/PBPA - Run 1
- · - ·	AMBC/PBPA - Run 5

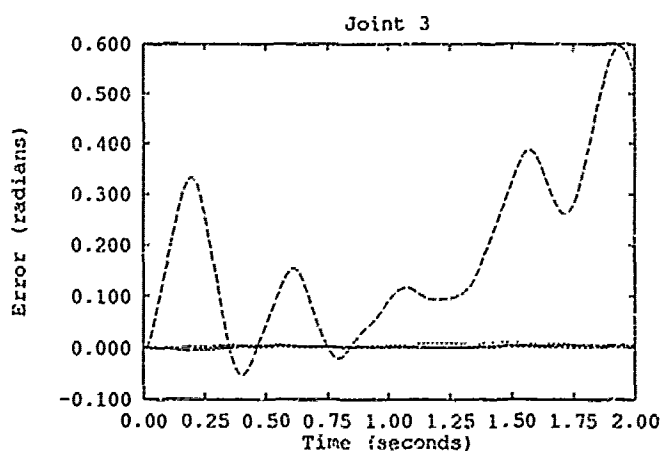
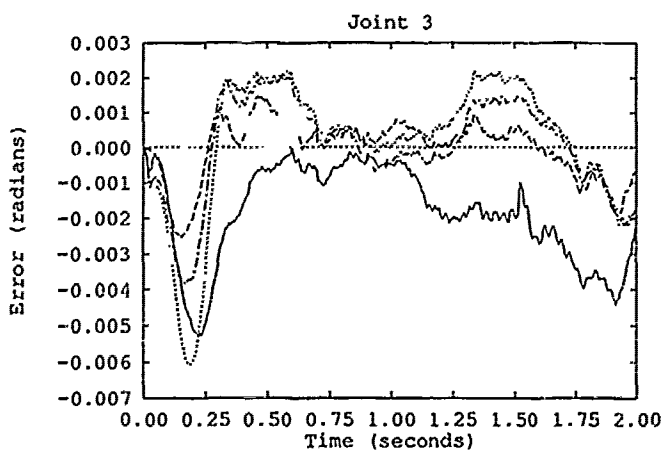
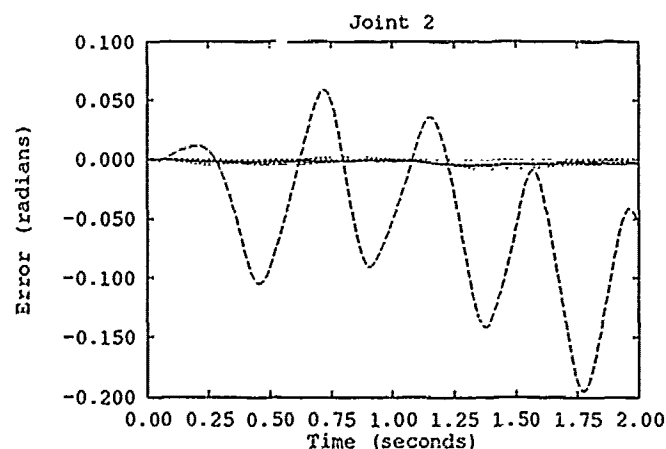
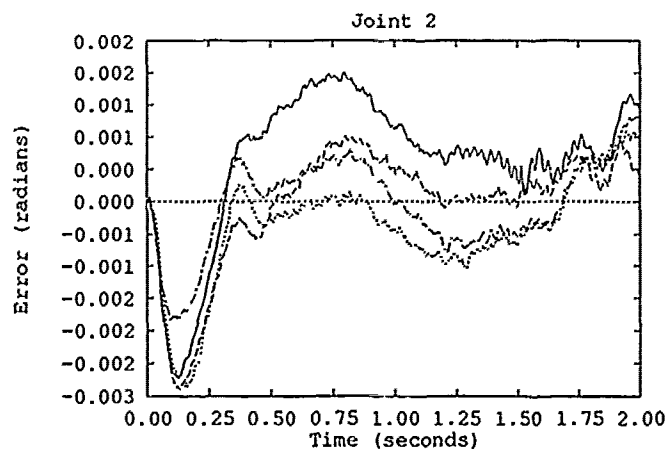
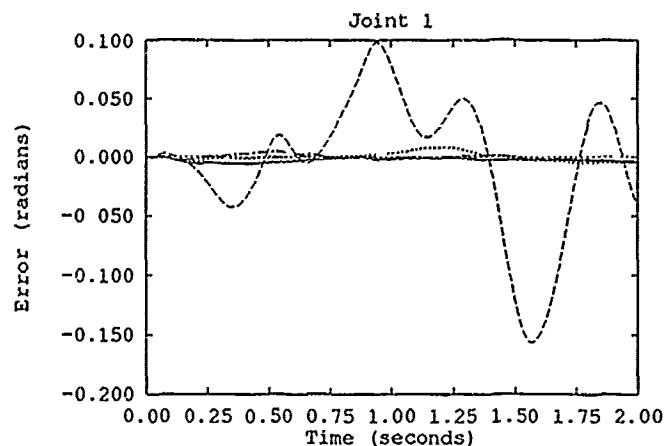
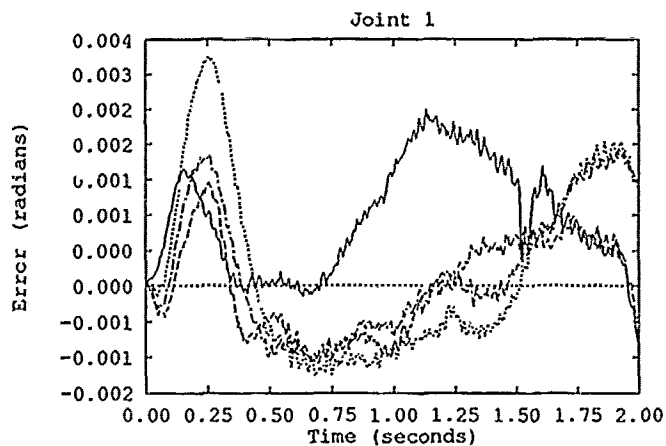


Figure A.5. AMBC/H vs. AMBC/PBPA - Trajectory 3

Figure A.6. AMBC/H vs. AMBC/PBPA - Trajectory 4

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
.....	AMBC/PBPA - Run 1
- · - · -	AMBC/PBPA - Run 5

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
.....	AMBC/PBPA - Run 1
- · - · -	AMBC/PBPA - Run 5

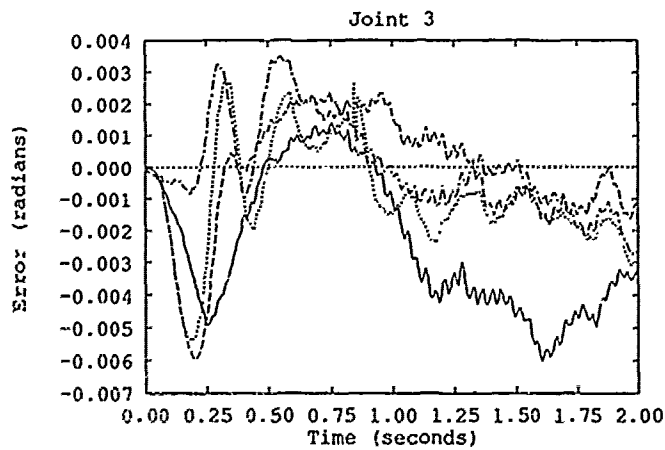
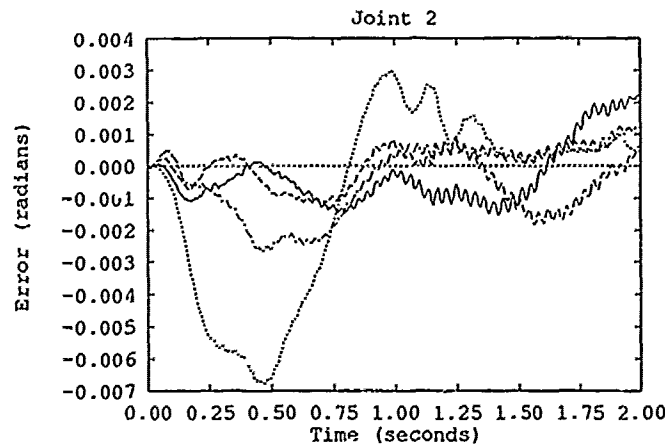
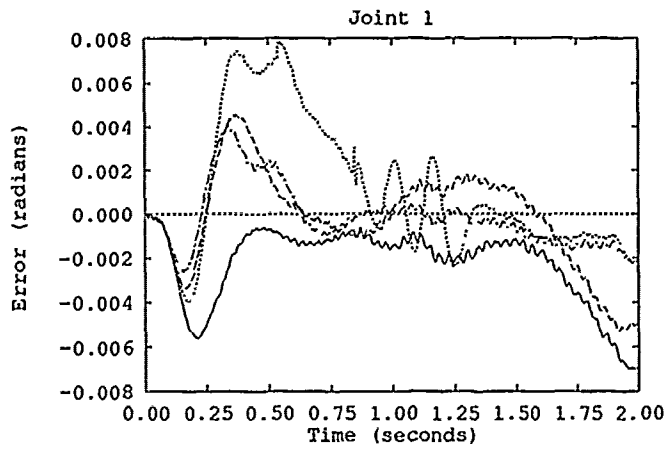


Figure A.7. AMBC/H vs. AMBC/PBPA - Trajectory 5

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
...	AMBC/PBPA - Run 1
- . - .	AMBC/PBPA - Run 5

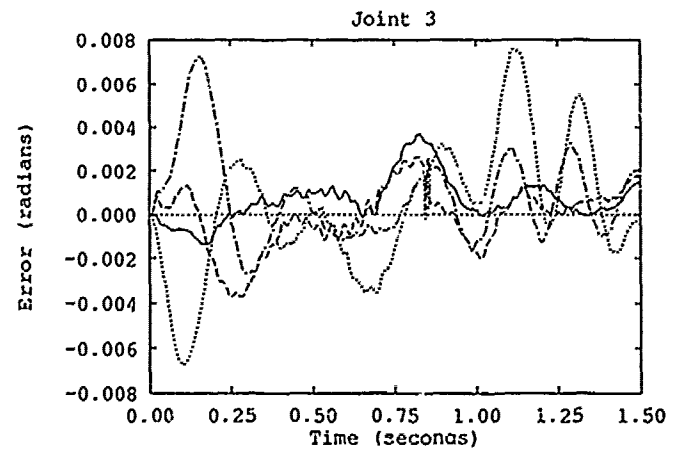
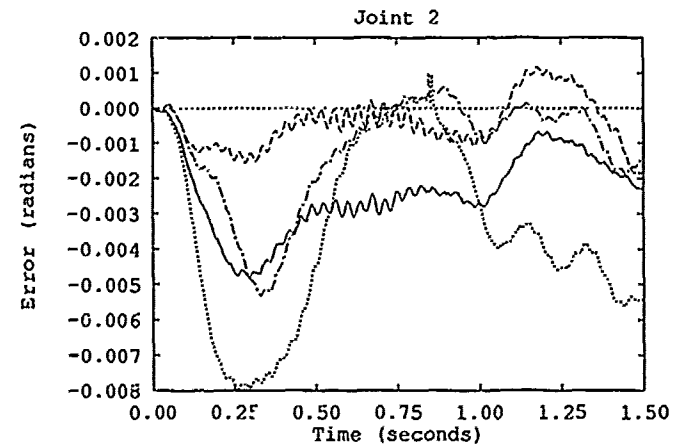
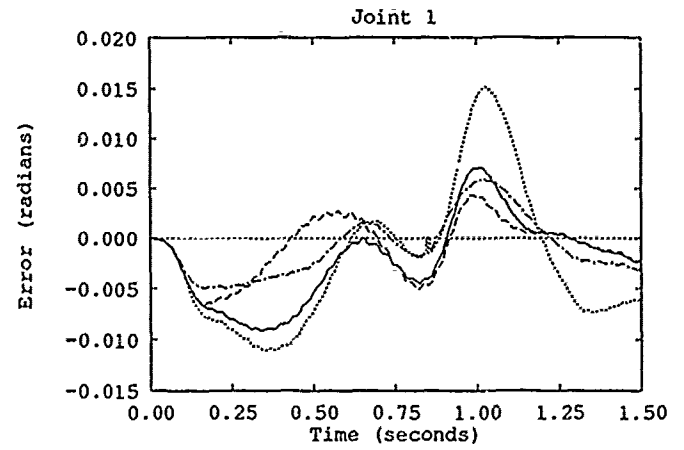


Figure A.8. AMBC/H vs. AMBC/PBPA - Trajectory 6

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
...	AMBC/PBPA - Run 1
- . - .	AMBC/PBPA - Run 5

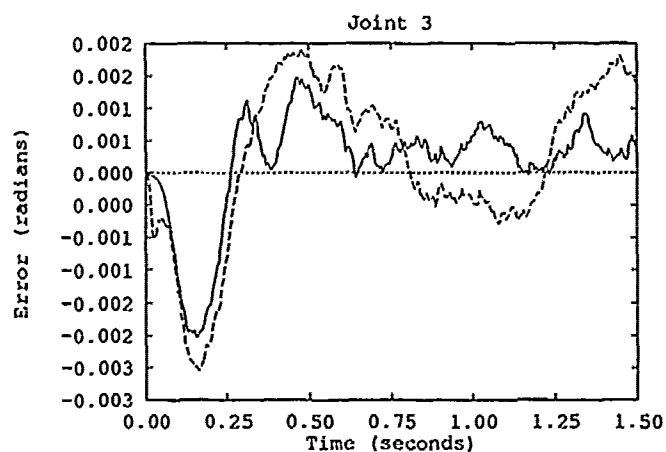
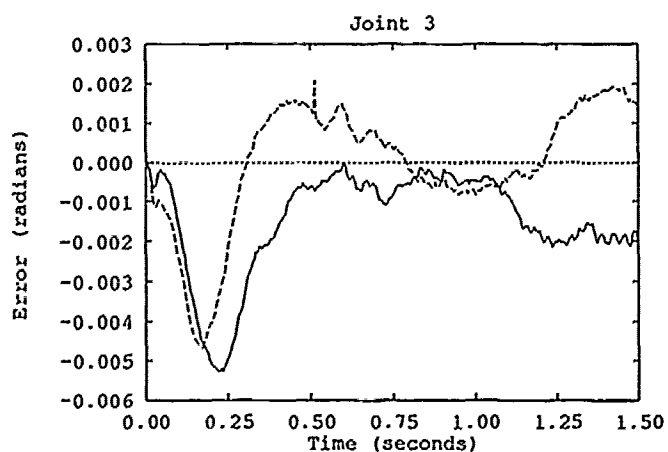
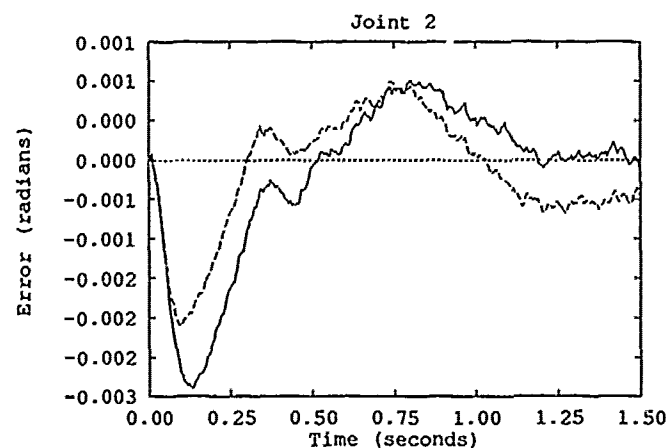
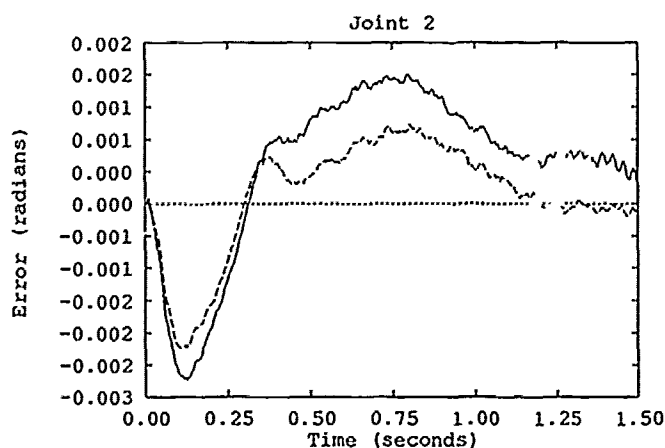
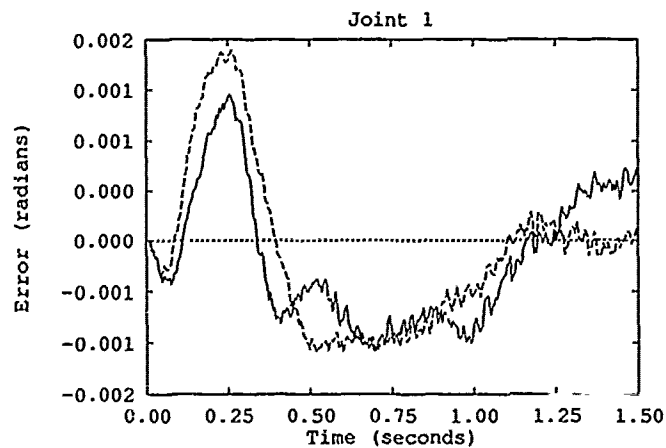
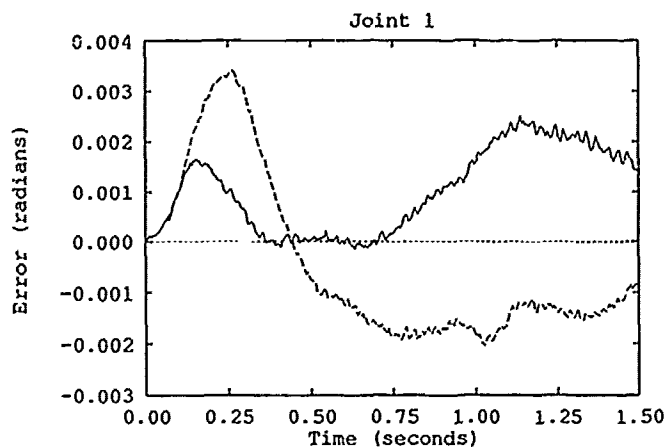


Figure A.9. Demonstration of Whether or Not AMBC/H and AMBC/PBPA Ever Become Equal: Run 1, Traj 3

—	AMBC/H
- - -	AMBC/PBPA

Figure A.10. Demonstration of Whether or Not AMBC/H and AMBC/PBPA Ever Become Equal: Run 5, Traj 3

—	AMBC/H
- - -	AMBC/PBPA

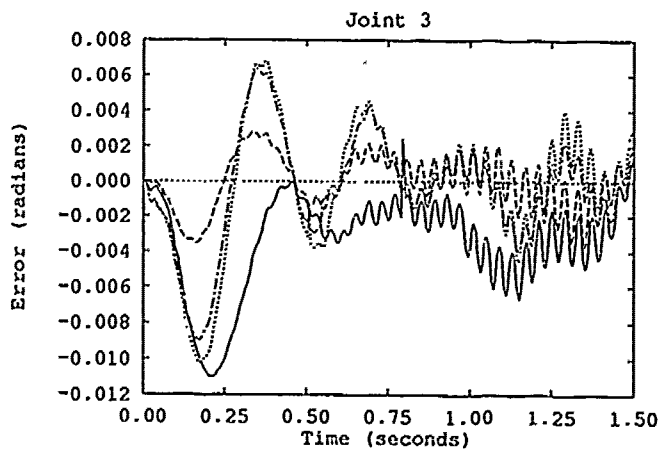
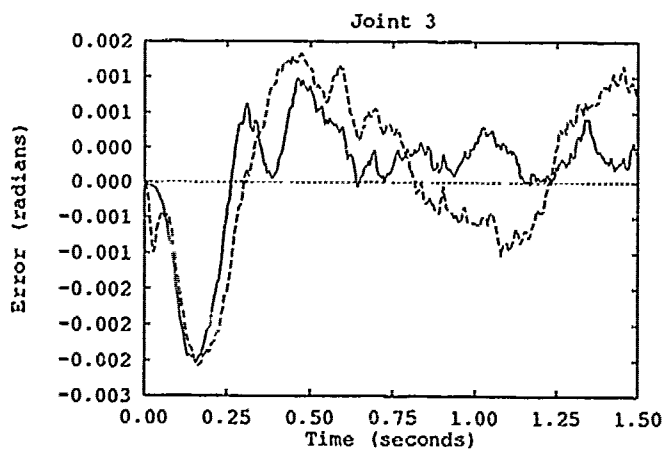
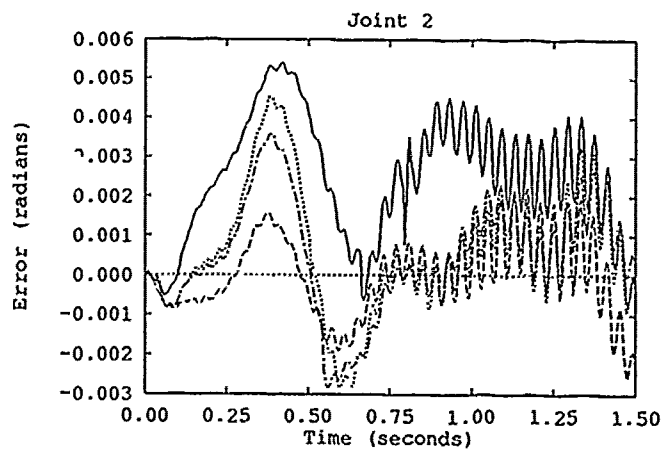
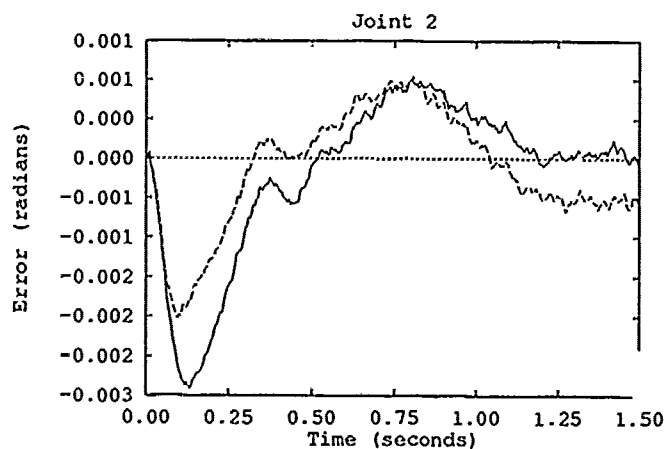
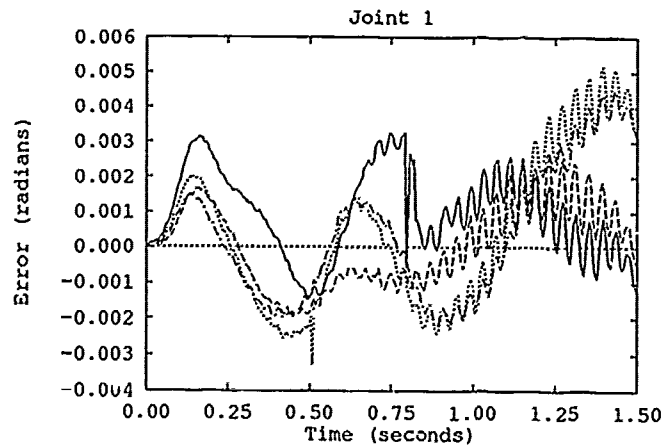
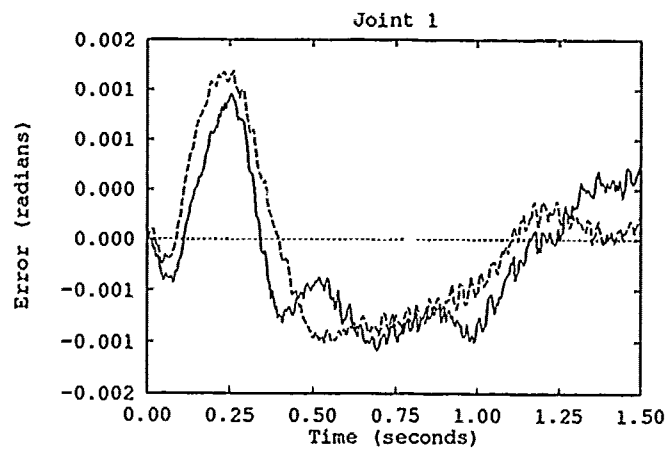


Figure A.11. Demonstration of Whether or Not AMBC/H and AMBC/PBPA Ever Become Equal, Traj 3

—	AMBC/H: Run 5
- - -	AMBC/PBPA: Run 10

Figure A.12. AMBC/H vs. AMBC/PBPA - Trajectory 1 with 2KG Payload

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
.....	AMBC/PBPA - Run 1
- . - . -	AMBC/PBPA - Run 5

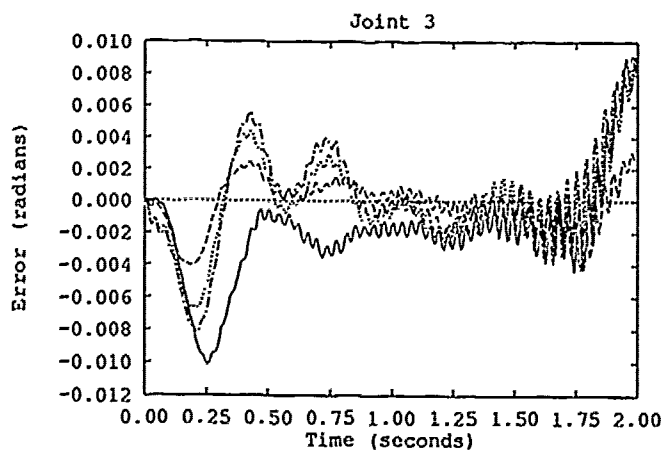
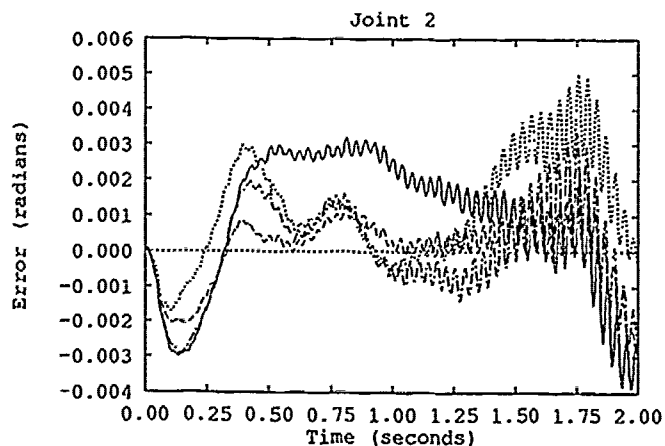
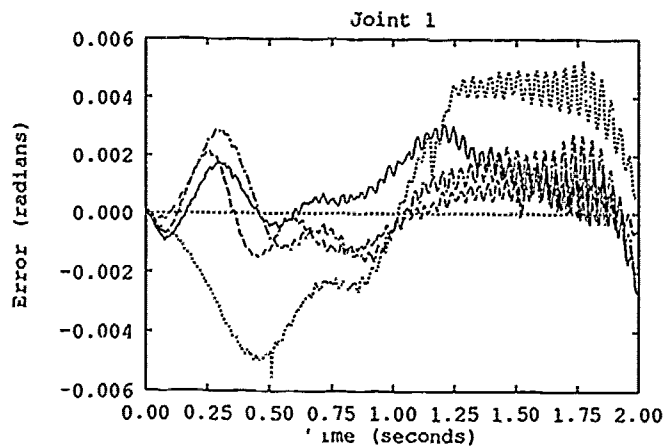


Figure A.13. AMBC/H vs. AMBC/PBPA - Trajectory 3 with 2KG Payload

—	AMBC/H - Run 1
- - -	AMBC/H - Run 5
...	AMBC/PBPA - Run 1
- · - ·	AMBC/PBPA - Run 5

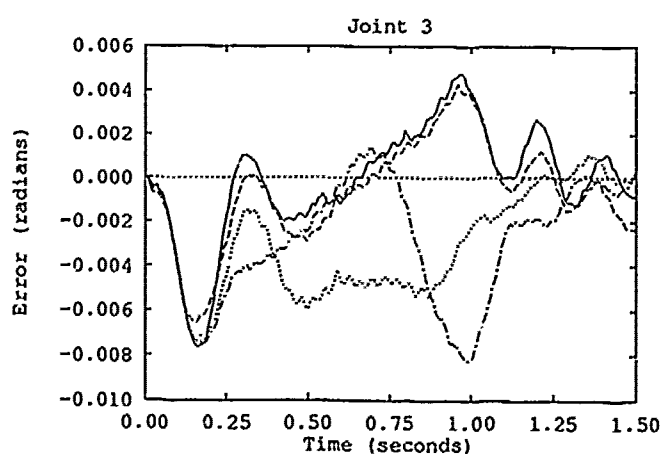
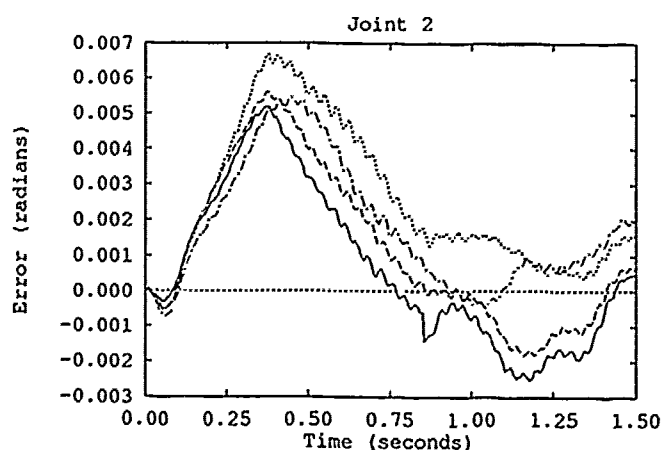
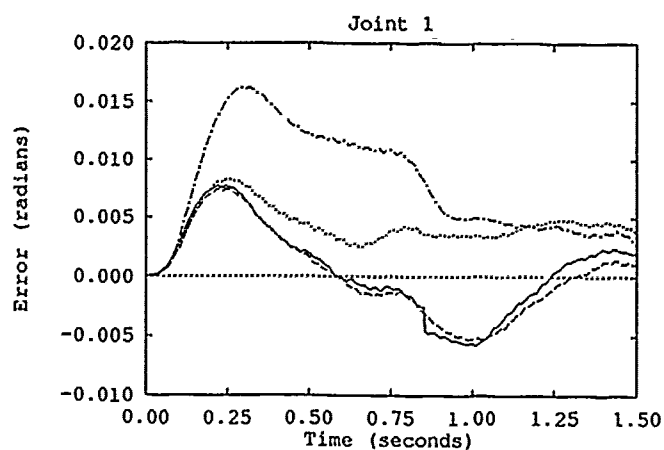


Figure A.14. Effect of Parameter Reduction: First Run, Trajectory 1

—	16 Parameters
- - -	13 Parameters
...	10 Parameters
- · - ·	7 Parameters

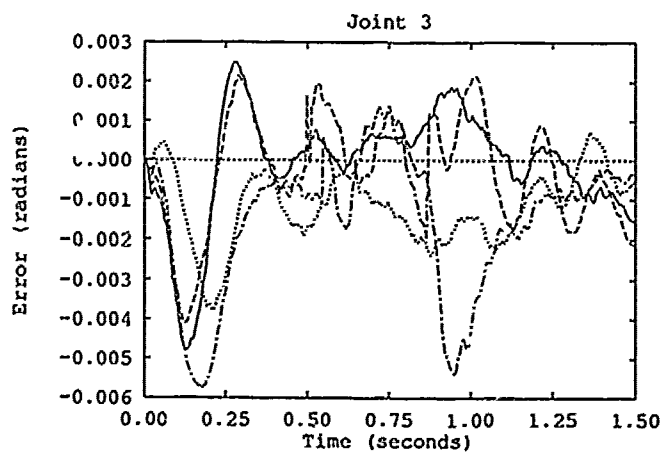
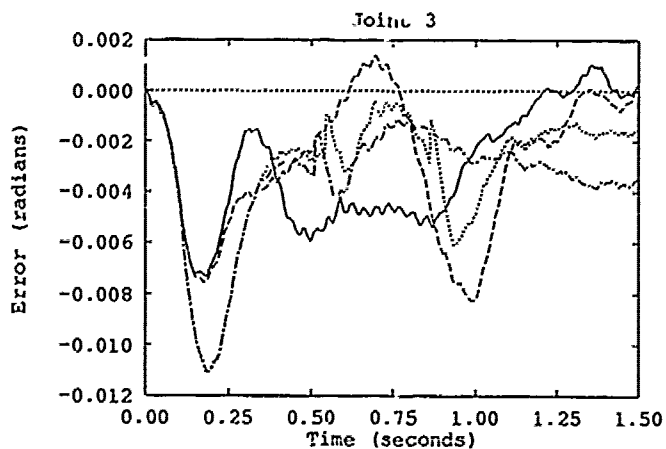
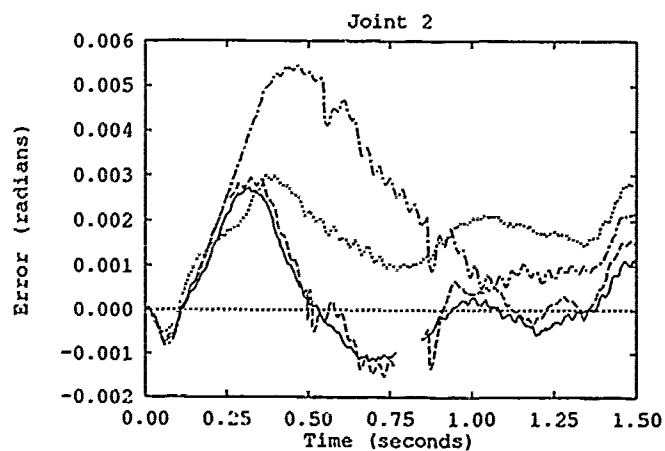
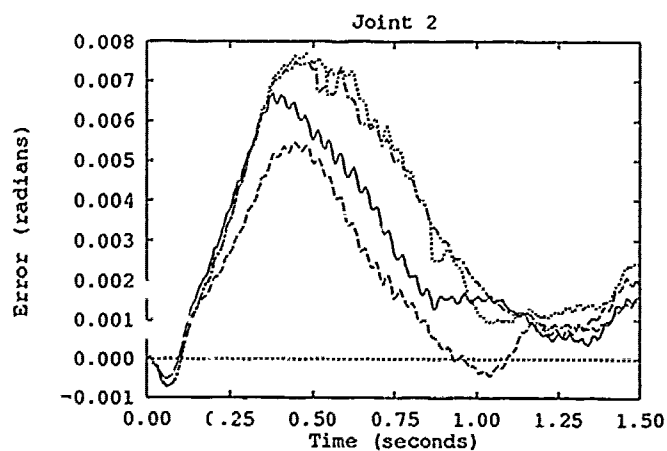
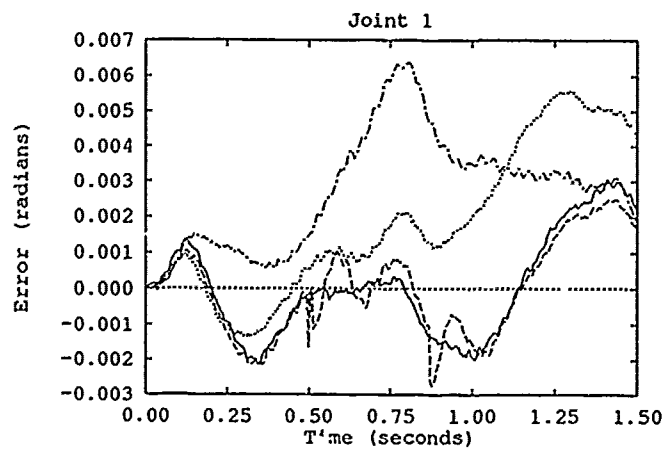
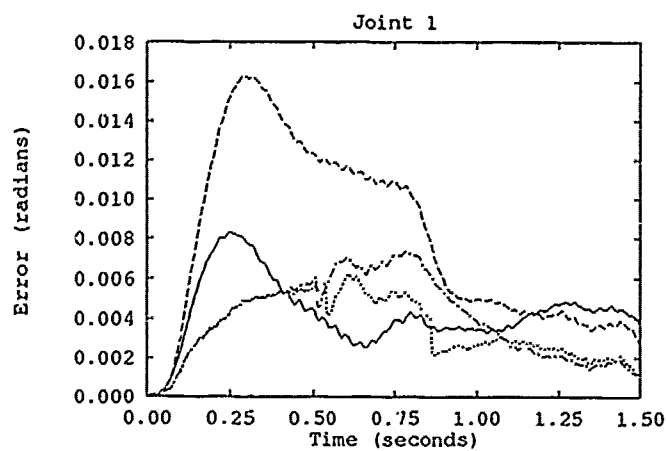


Figure A.15. Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 1

—	10 Parameters
- - -	7 Parameters
...	4 Parameters
- . - .	2 Parameters

Figure A.16. Effect of Parameter Reduction: Fifth Run, Trajectory 1

—	16 Parameters
- - -	13 Parameters
...	10 Parameters
- . - .	7 Parameters

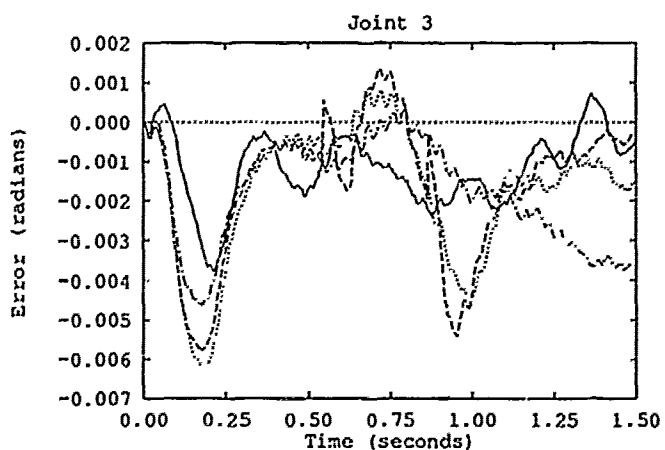
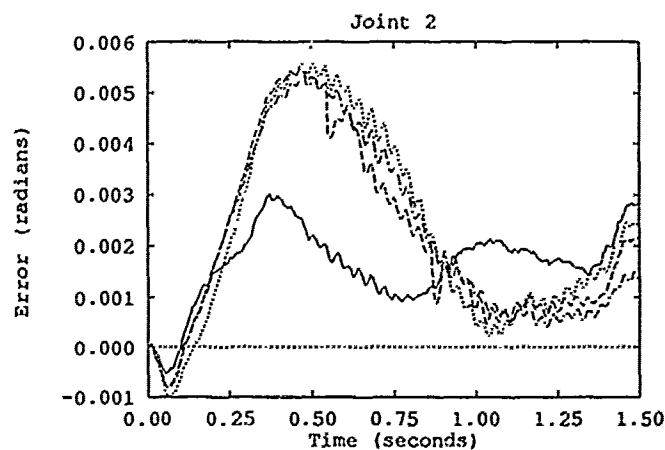
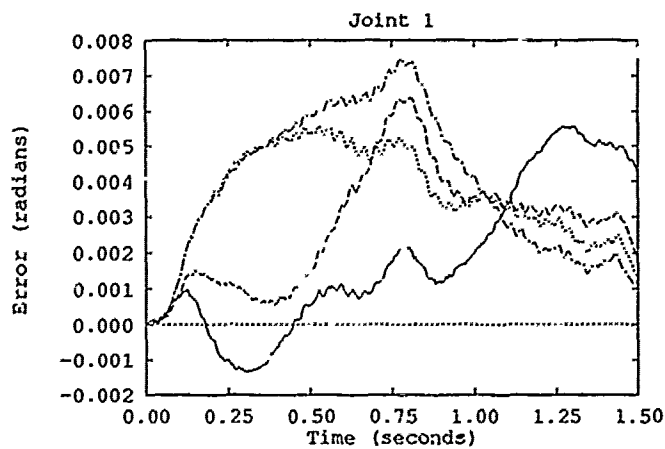


Figure A.17. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 1

—	10 Parameters
- - -	7 Parameters
...	4 Parameters
- . - .	2 Parameters

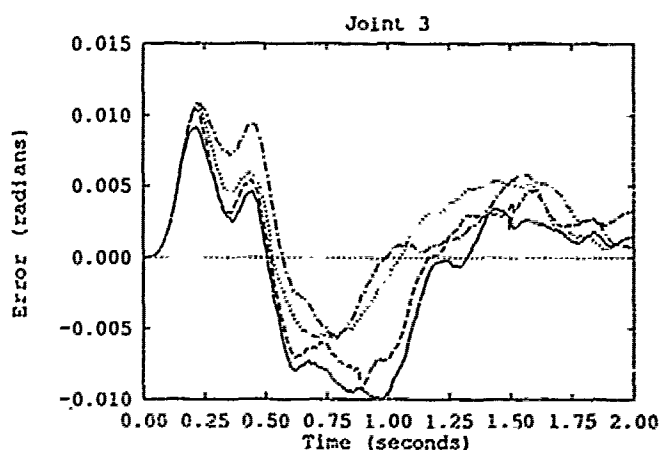
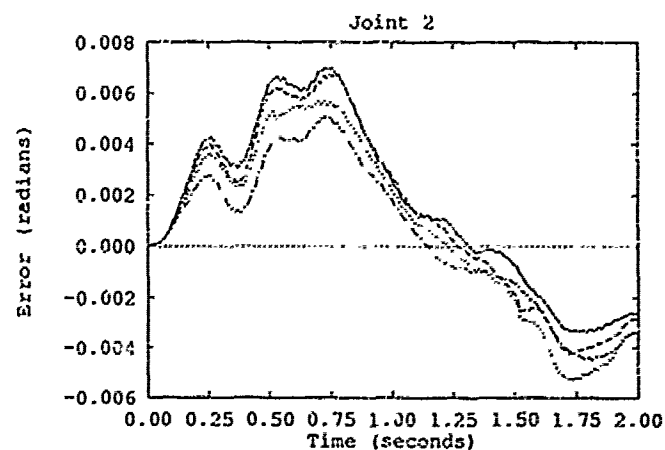
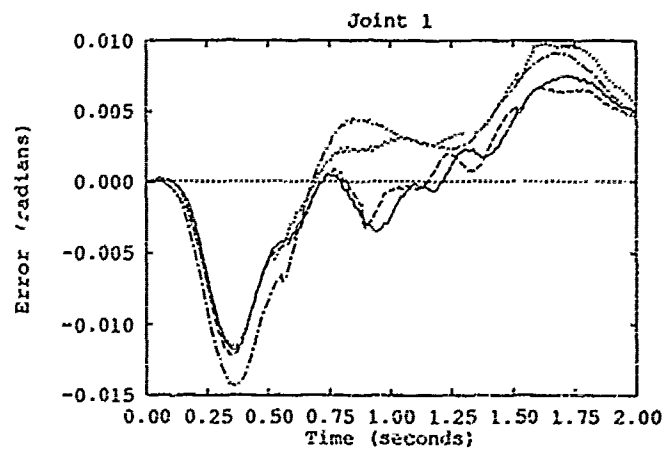


Figure A.18. Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 2

—	16 Parameters
- - -	13 Parameters
...	10 Parameters
- . - .	7 Parameters

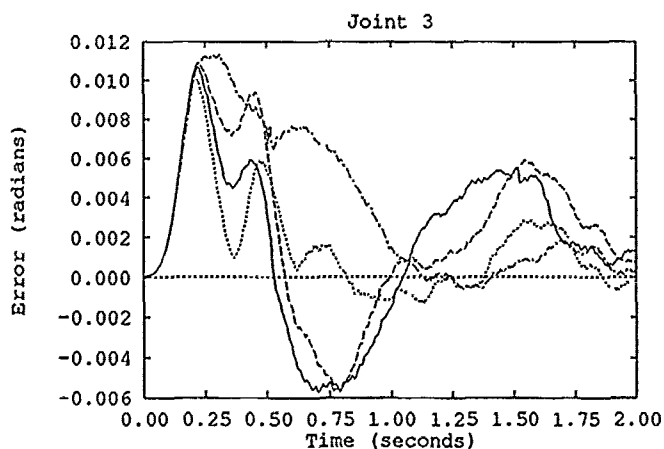
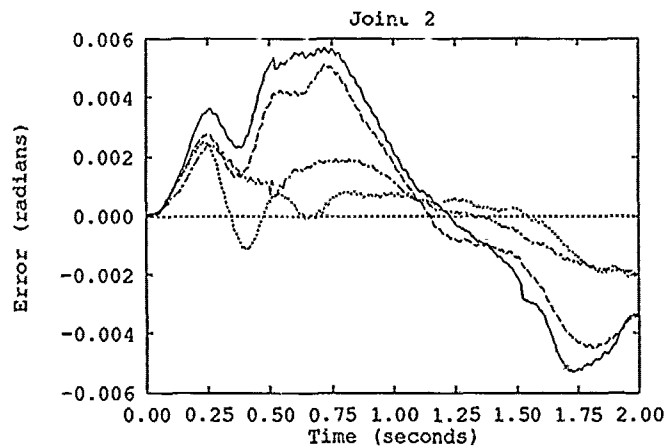
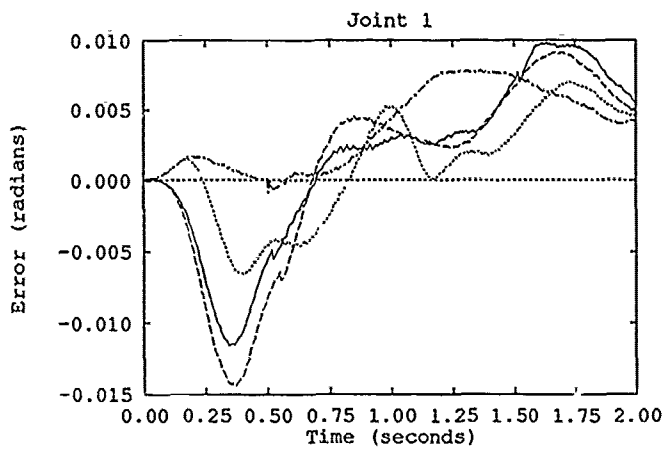


Figure A.19. Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 2

—	10 Parameters
- - -	7 Parameters
...	4 Parameters
- . - .	2 Parameters

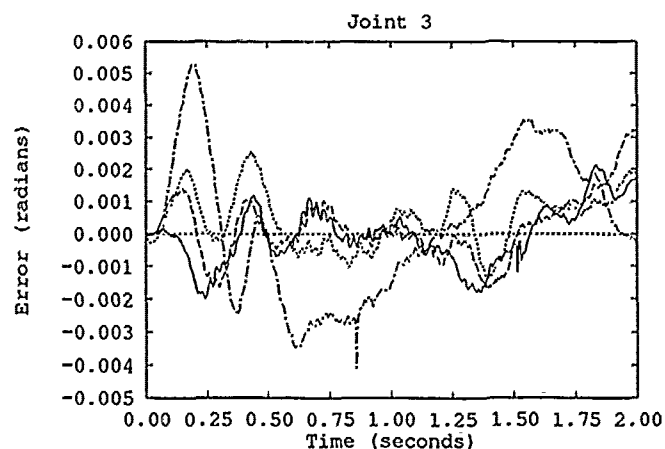
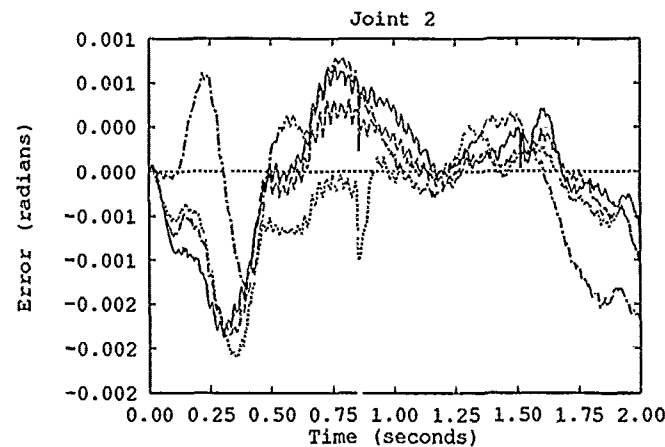
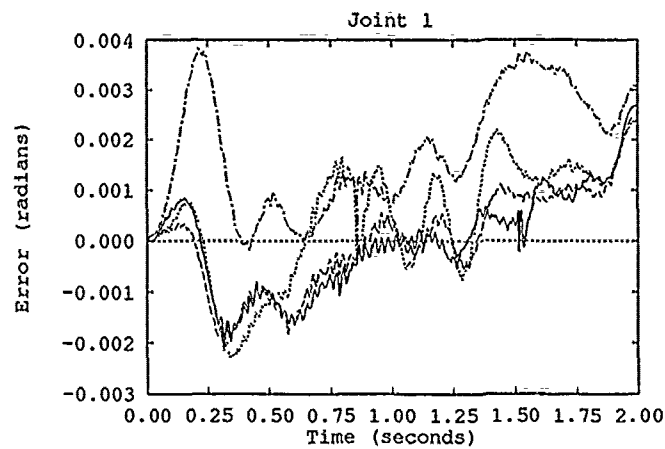


Figure A.20. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2

—	16 Parameters
- - -	13 Parameters
...	10 Parameters
- . - .	7 Parameters

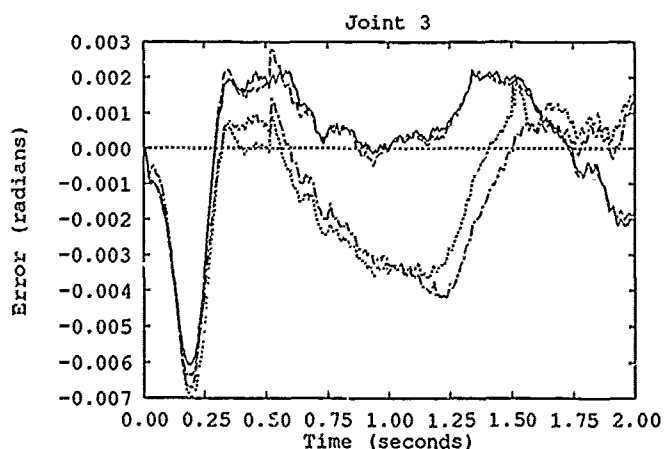
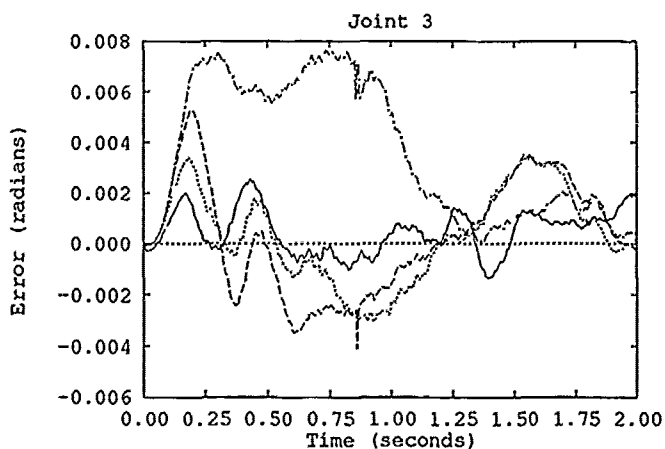
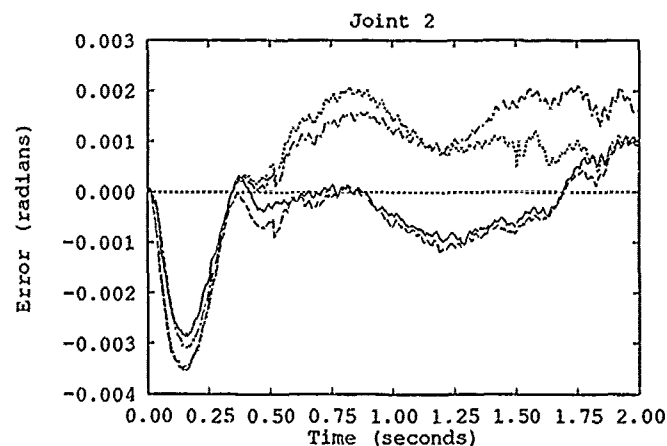
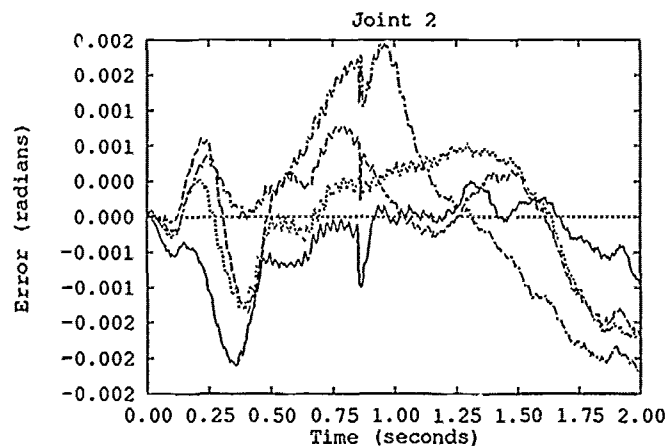
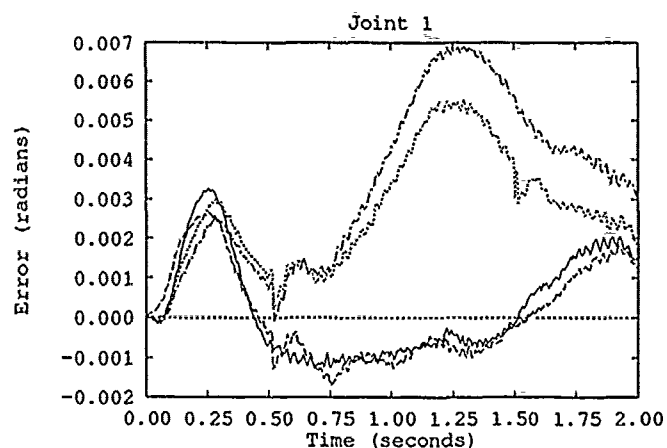
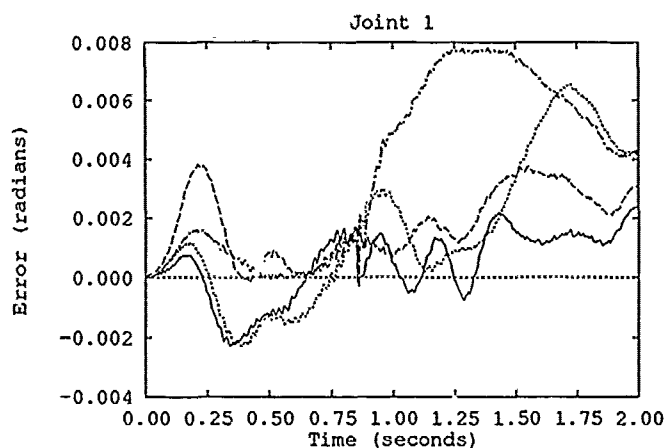


Figure A.21. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 2

—	10 Parameters
- - -	7 Parameters
· · ·	4 Parameters
- · - ·	2 Parameters

Figure A.22. Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 3

—	16 Parameters
- - -	13 Parameters
· · ·	10 Parameters
- · - ·	7 Parameters

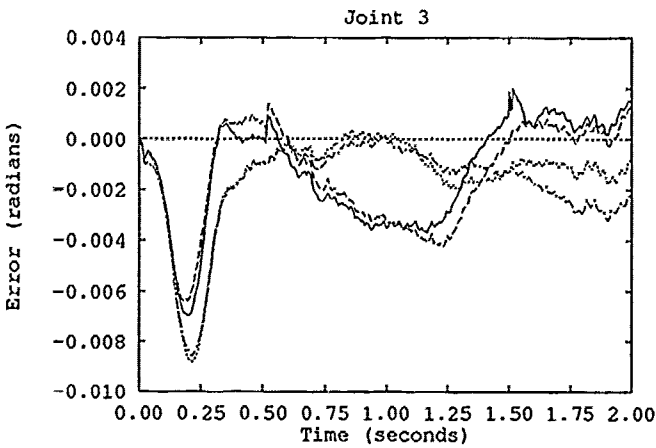
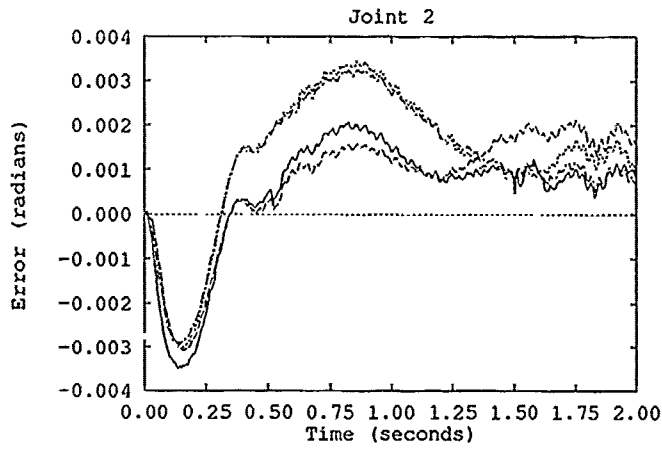
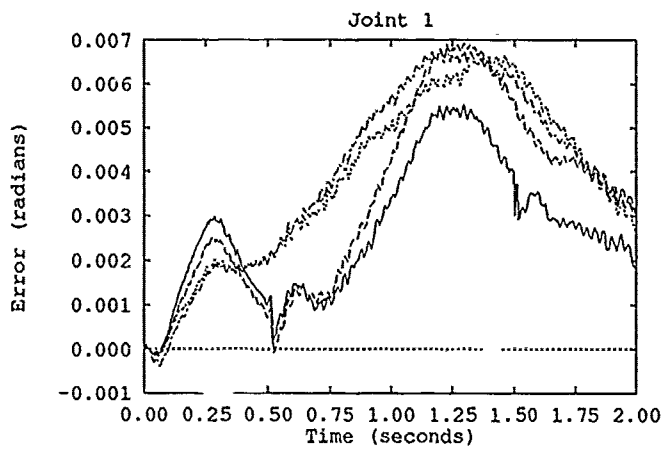


Figure A.23. Effect of Reducing Number of Estimated PBPA Parameters: First Run, Trajectory 3

—	10 Parameters
- - -	7 Parameters
.....	4 Parameters
- · - ·	2 Parameters

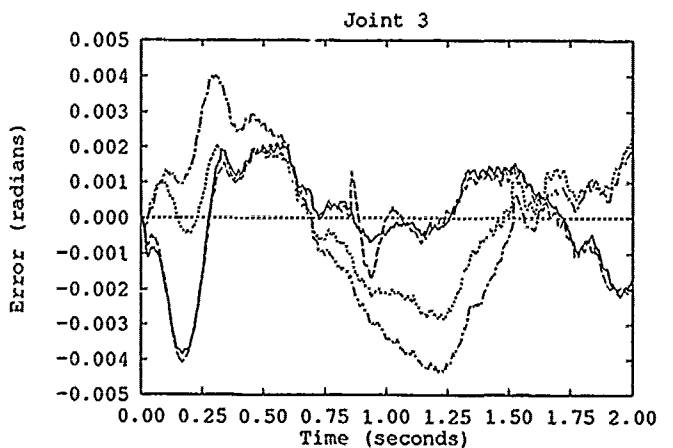
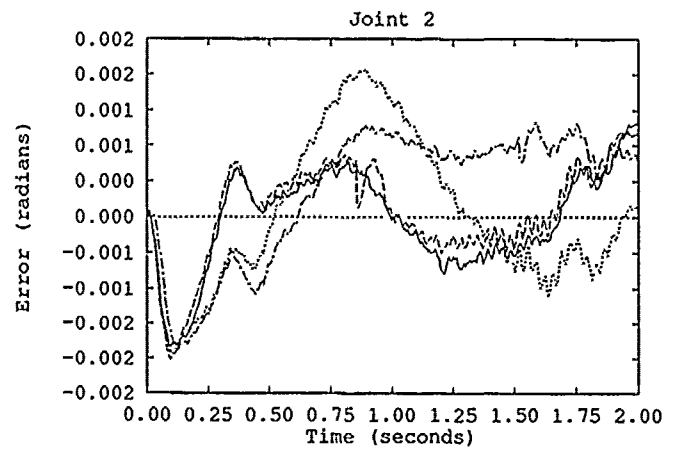
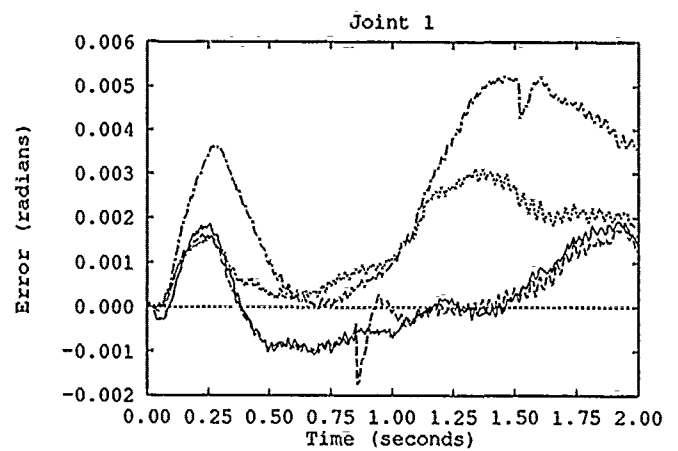


Figure A.24. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3

—	16 Parameters
- - -	13 Parameters
.....	10 Parameters
- · - ·	7 Parameters

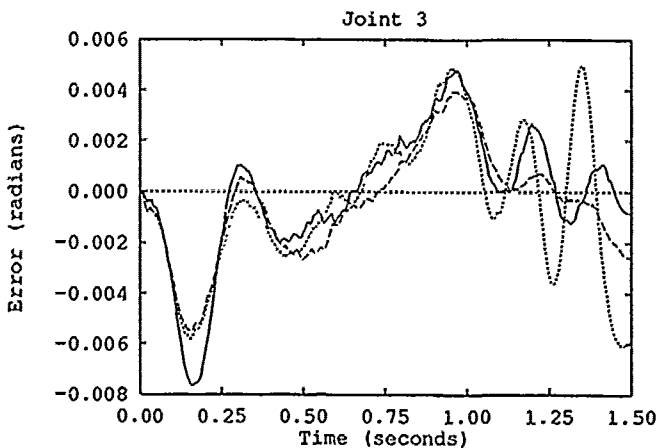
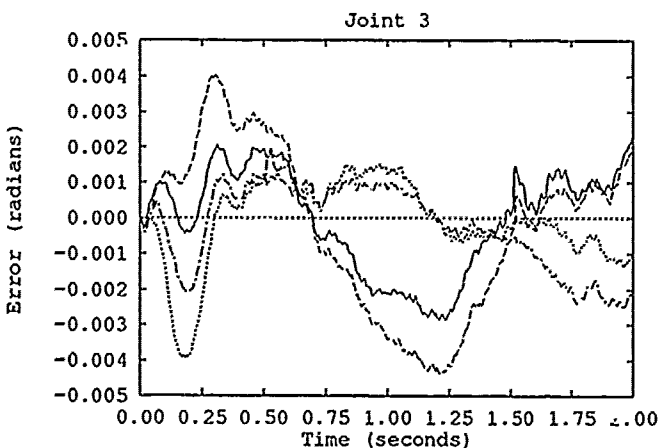
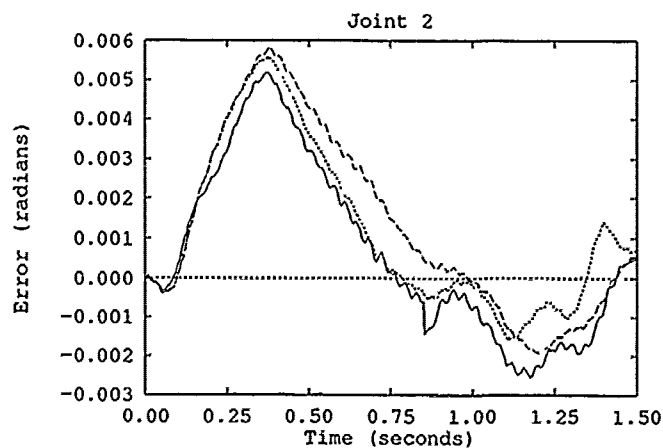
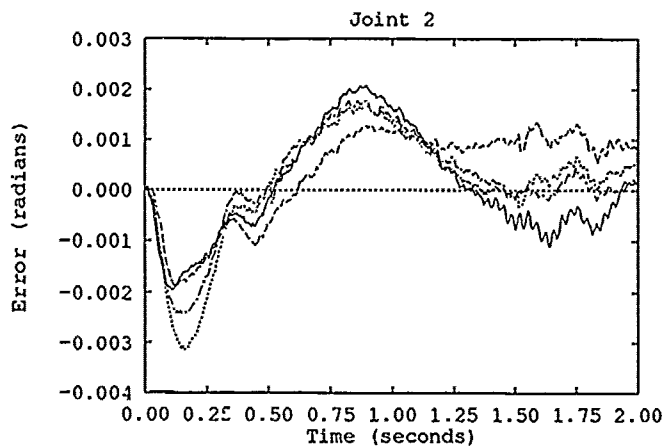
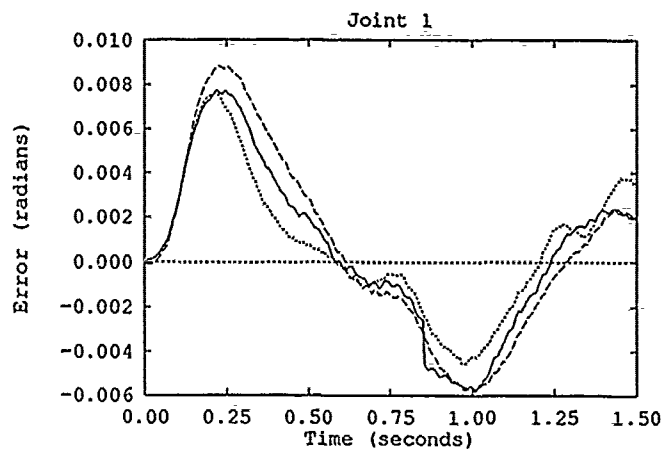
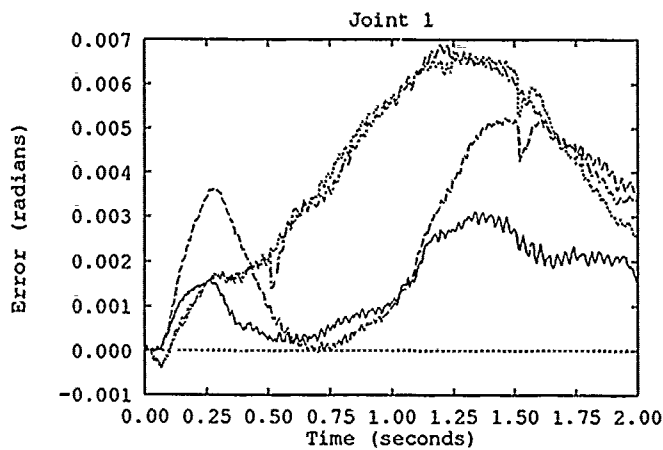


Figure A.25. Effect of Reducing Number of Estimated PBPA Parameters: Fifth Run, Trajectory 3

—	10 Parameters
- - -	7 Parameters
· · ·	4 Parameters
- · -	2 Parameters

Figure A.26. Effect of Varying Physical Parameter Values: Trajectory 1, Run 1

—	Best Estimate
- - -	Best Estimate +20%
· · ·	Best Estimate -20%

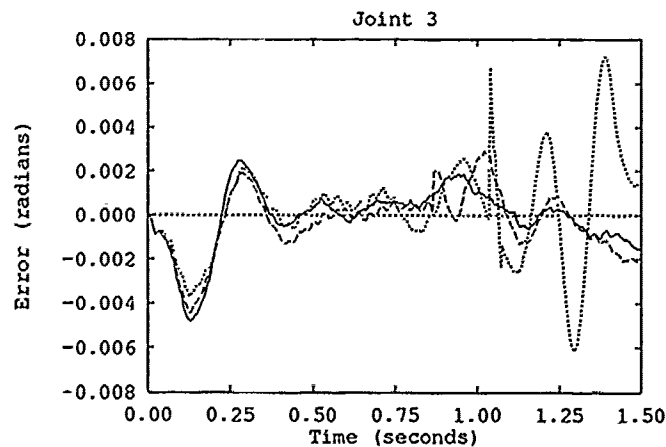
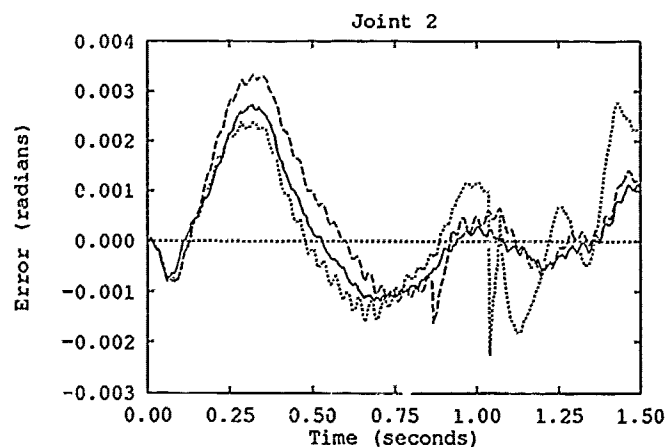
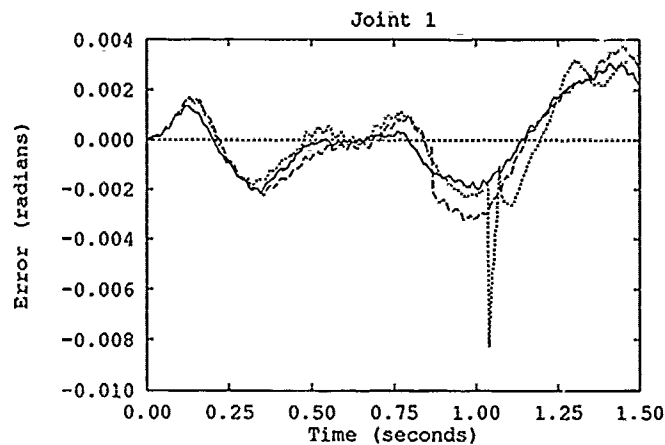


Figure A.27. Effect of Varying Physical Parameter Values: Trajectory 1, Run 5

	Best Estimate
	Best Estimate +20%
	Best Estimate -20%

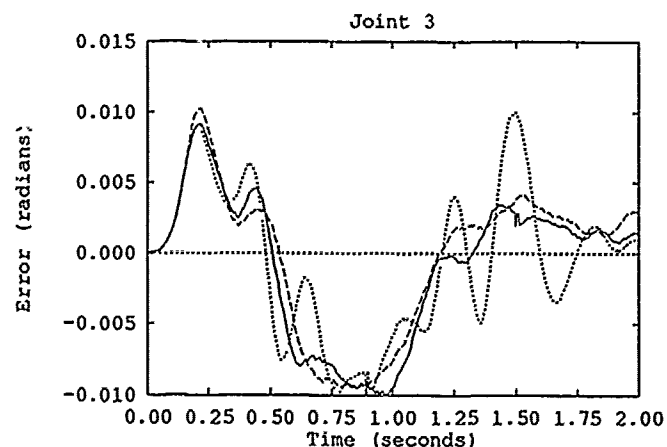
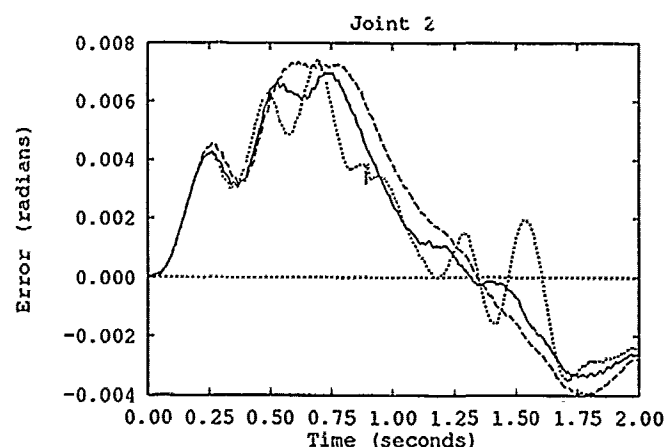
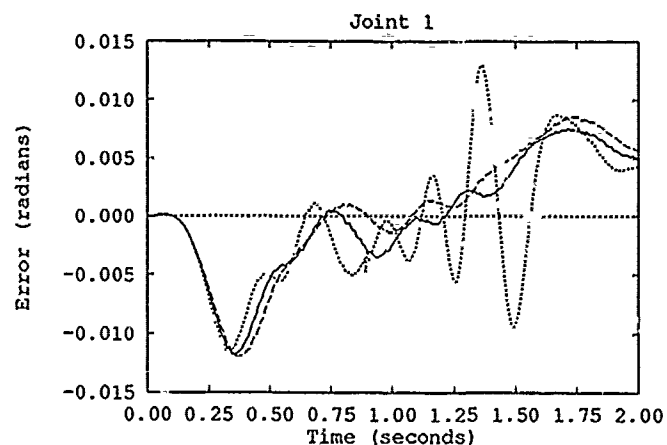


Figure A.28. Effect of Varying Physical Parameter Values: Trajectory 2, Run 1

	Best Estimate
	Best Estimate +20%
	Best Estimate -20%

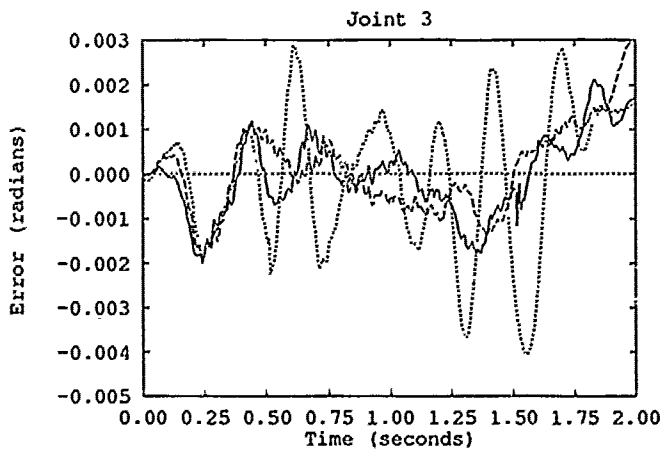
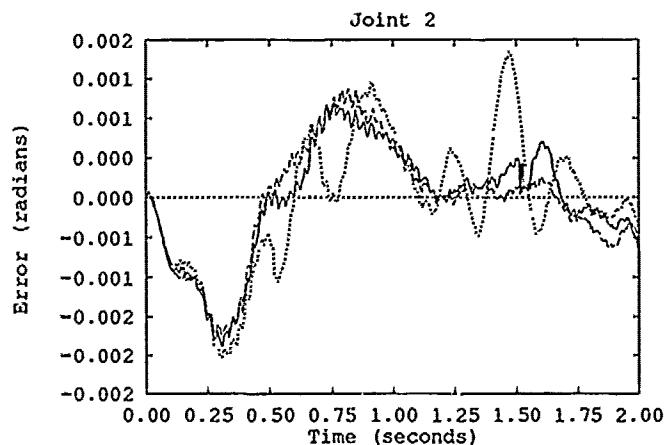
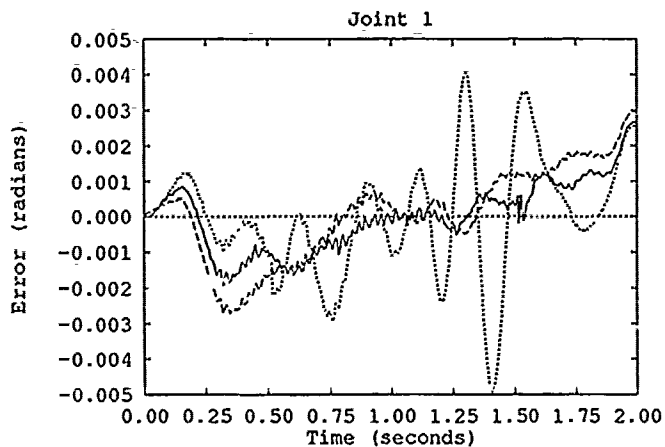


Figure A.29. Effect of Varying Physical Parameter Values: Trajectory 2, Run 5

—	Best Estimate
- - -	Best Estimate +20%
.....	Best Estimate -20%

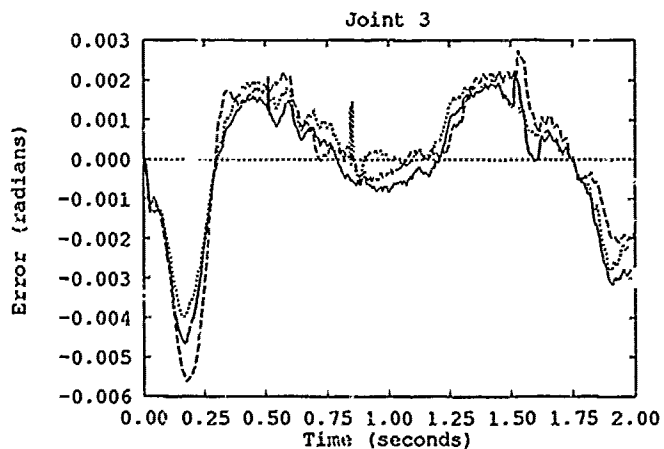
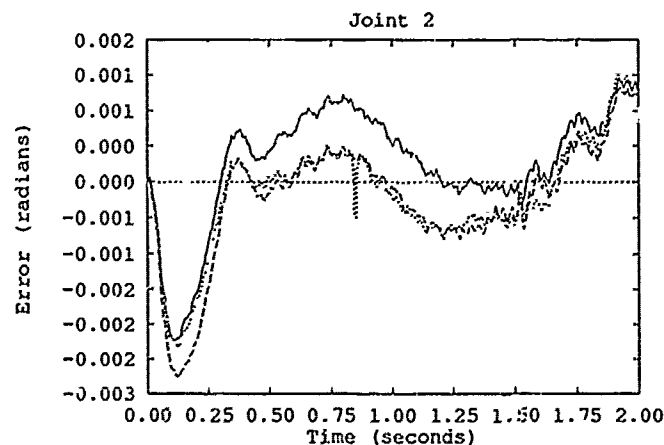
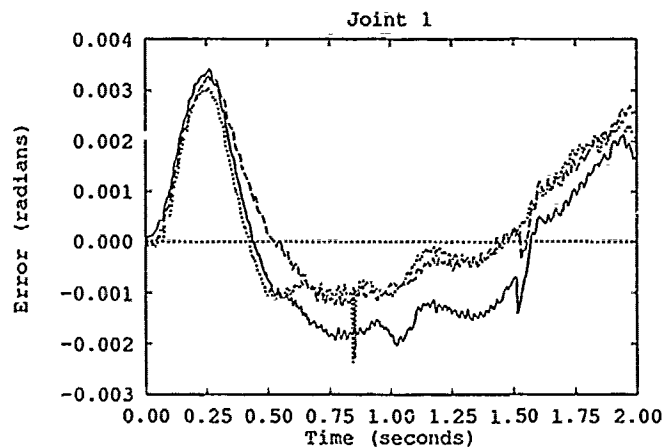


Figure A.30. Effect of Varying Physical Parameter Values: Trajectory 3, Run 1

—	Best Estimate
- - -	Best Estimate +20%
.....	Best Estimate -20%

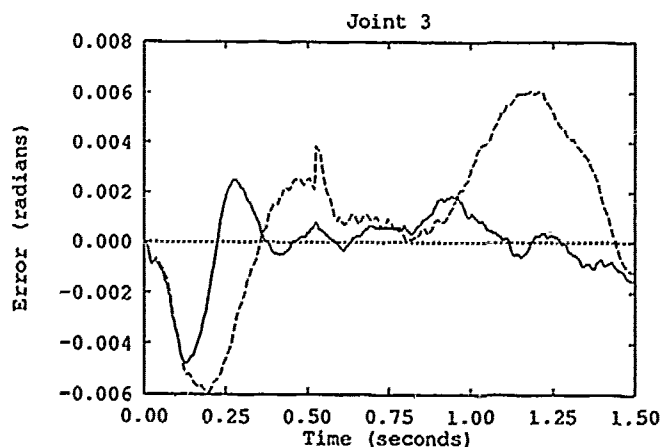
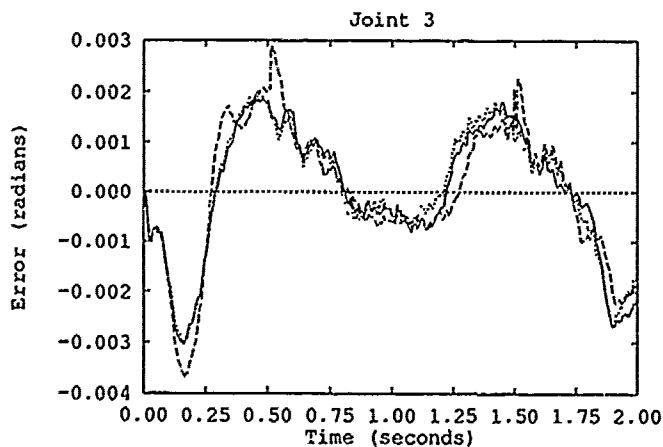
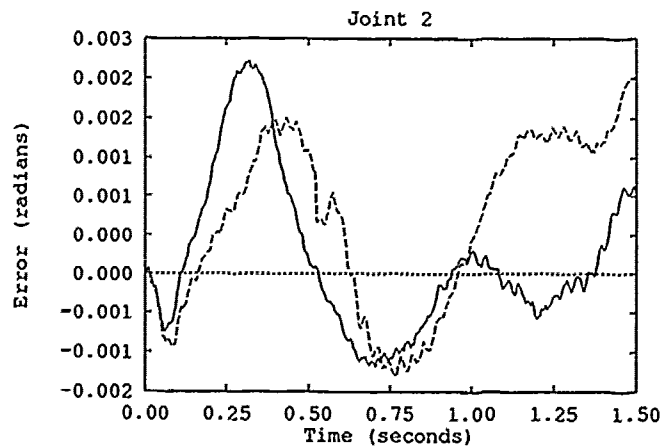
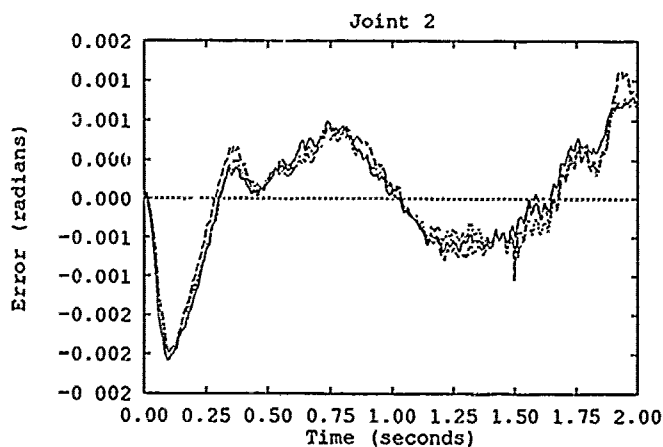
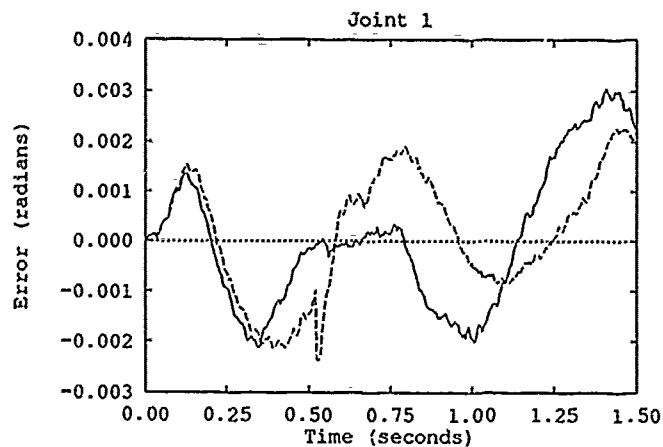
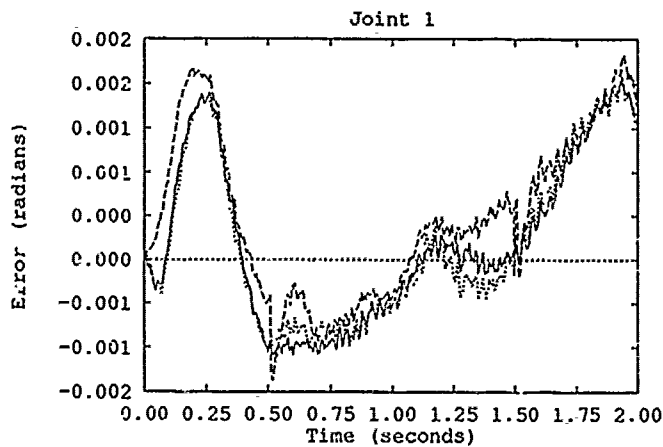


Figure A.31. Effect of Varying Physical Parameter Values: Trajectory 3, Run 5

—	Best Estimate
- - -	Best Estimate +20%
· · ·	Best Estimate -20%

Figure A.32. Effect of Discontinuing Adaptation - Trajectory 1

—	AMBC/PBPA with Adaptation
- - -	AMBC/PBPA without Adaptation

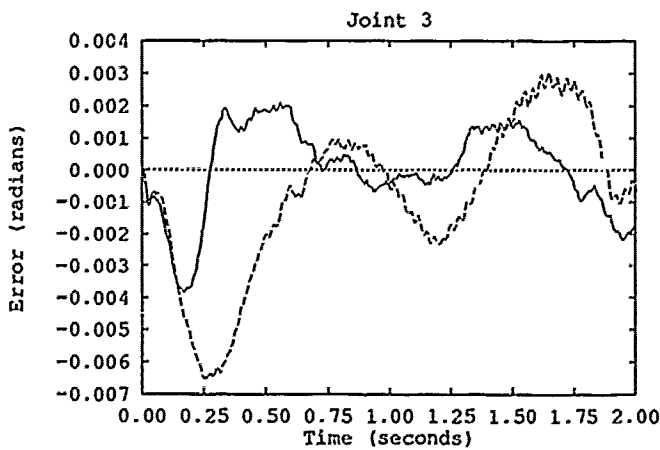
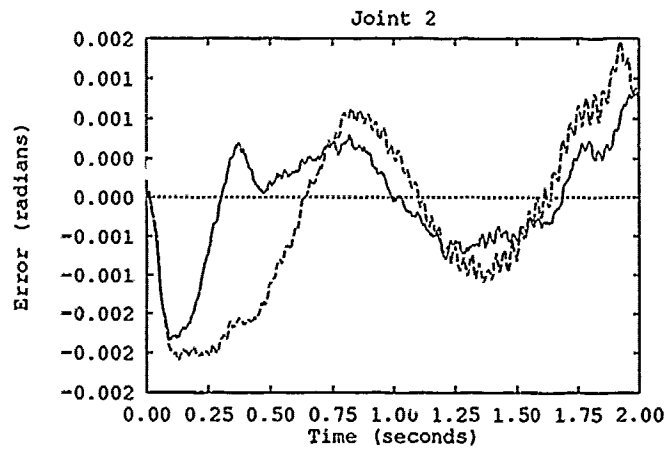
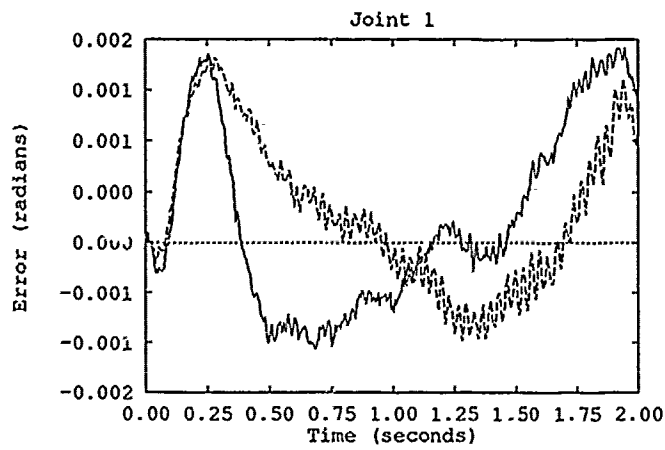


Figure A.33. Effect of Discontinuing Adaptation - Trajectory 3

—	AMBC/PBPA with Adaptation
- - -	AMBC/PBPA without Adaptation

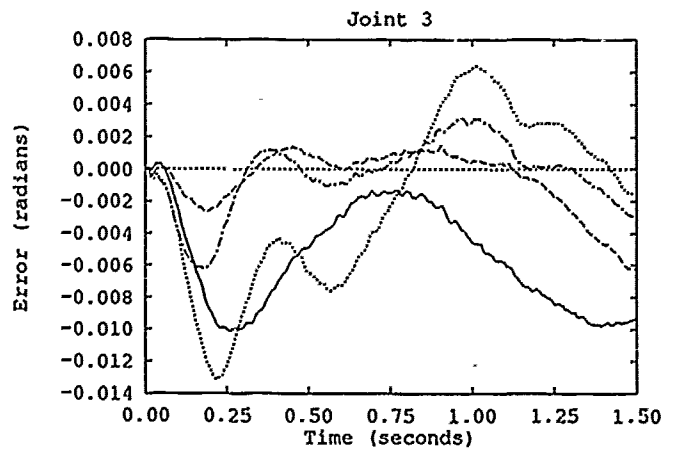
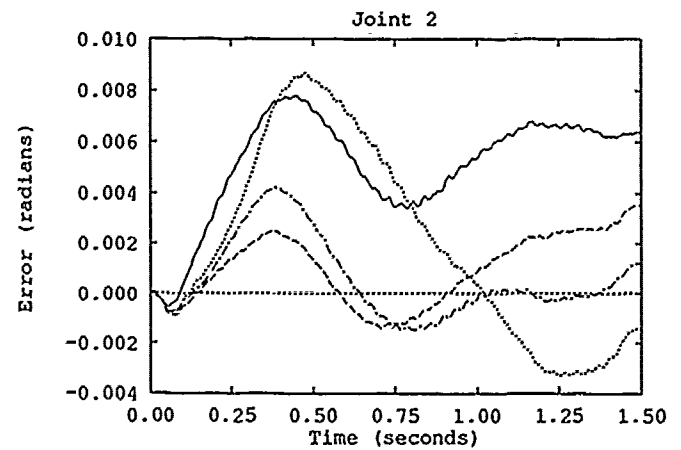
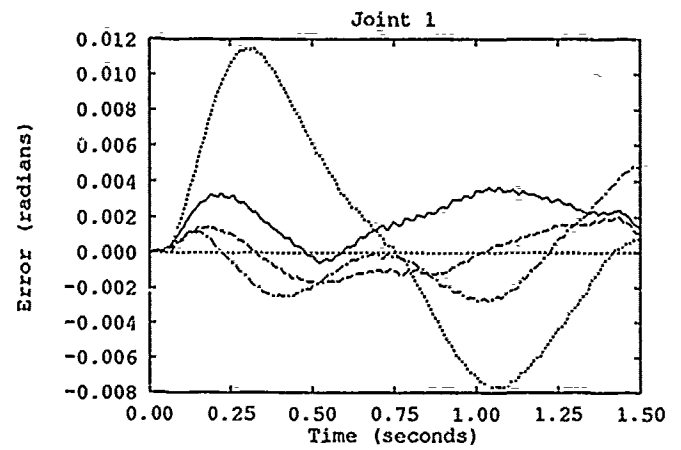


Figure A.34. Effect of Soft PD Gains, Trajectory 1, No Payload

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
.....	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 5

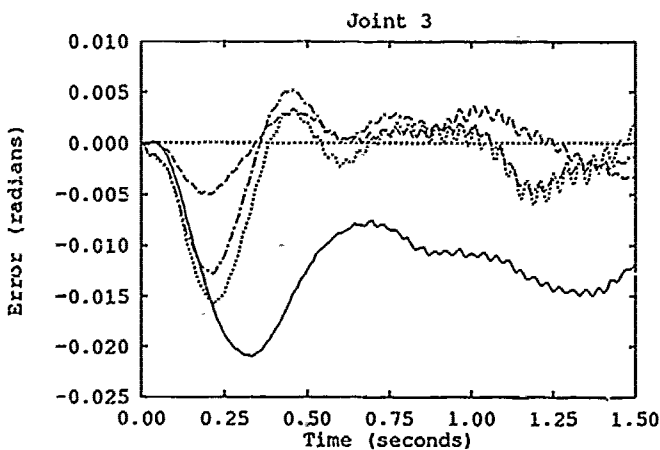
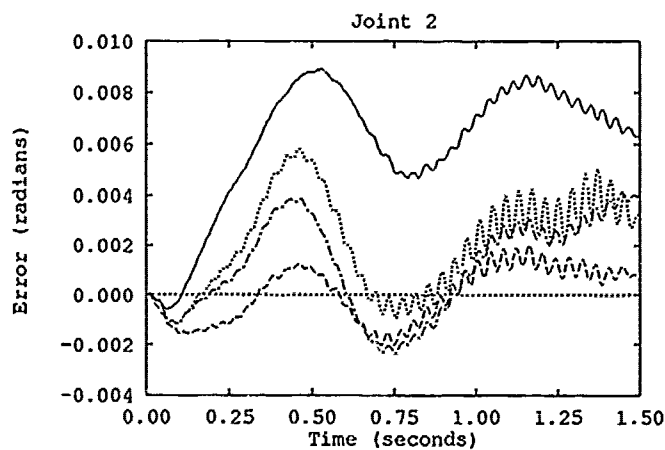
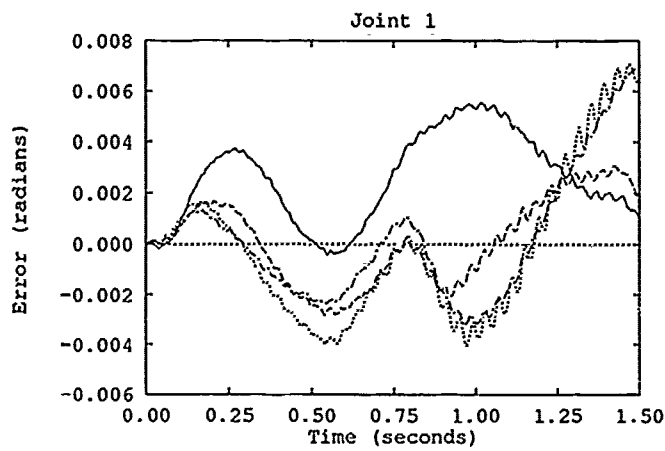


Figure A.35. Effect of Soft PD Gains, Trajectory 1, 2KG Payload

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
...	AMBC/PBPA, Run 1
- · - ·	AMBC/PBPA, Run 5

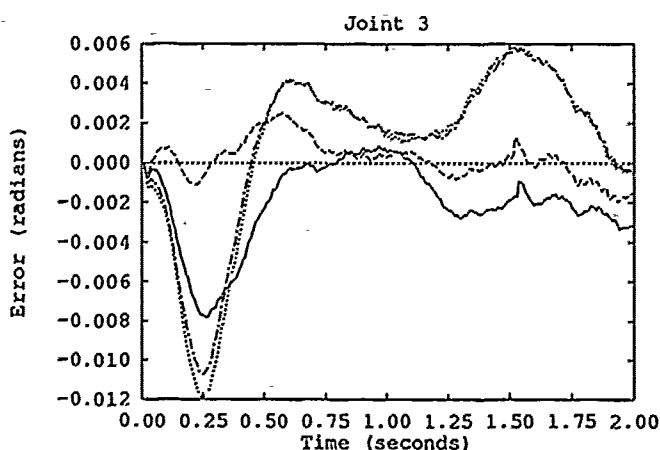
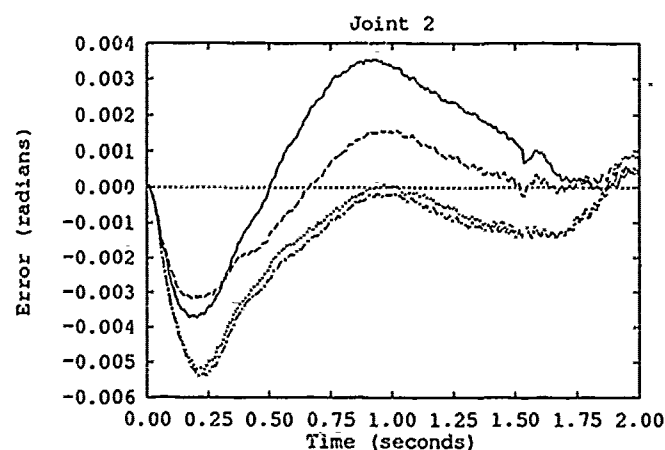
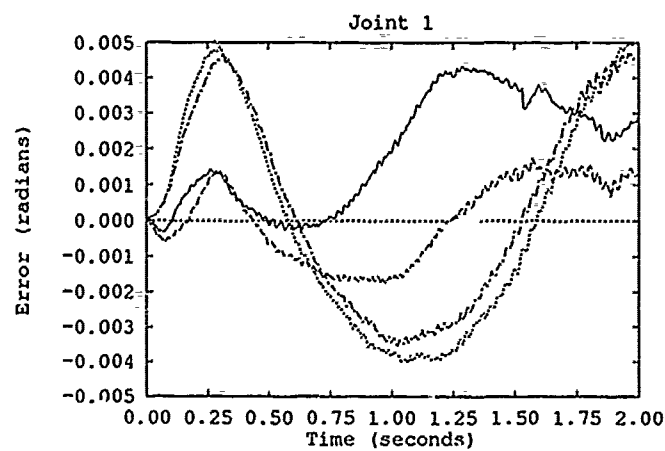


Figure A.36. Effect of Soft PD Gains, Trajectory 3, No Payload

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
...	AMBC/PBPA, Run 1
- · - ·	AMBC/PBPA, Run 5

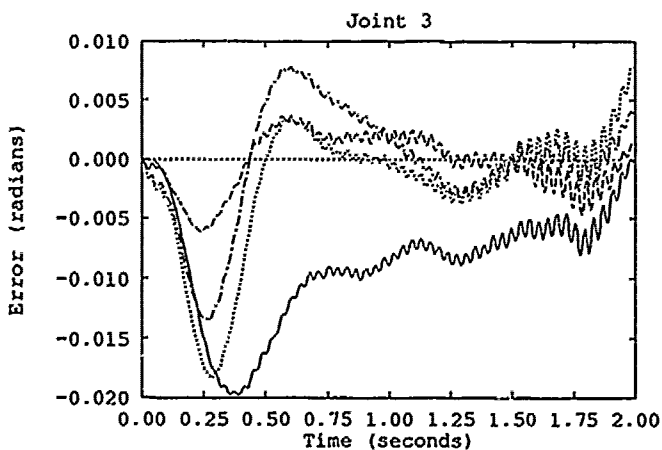
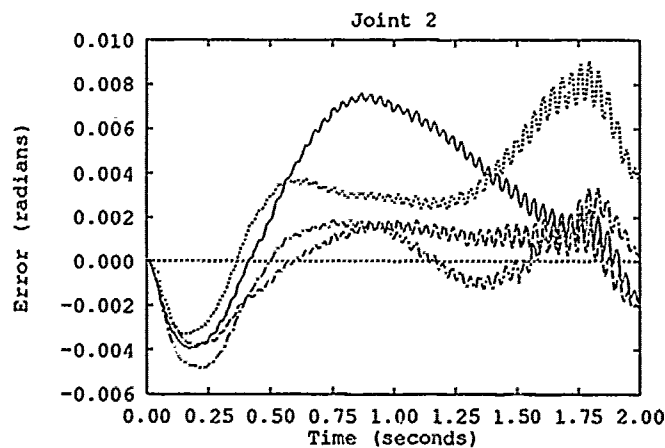
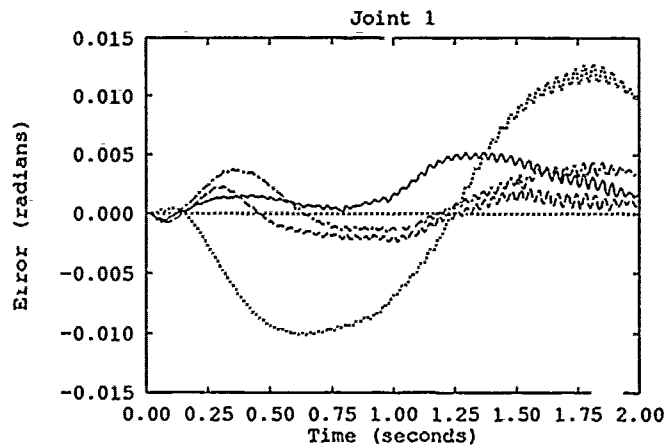


Figure A.37. Effect of Soft PD Gains, Trajectory 3, 2KG Payload

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
...	AMBC/PBPA, Run 1
- . - .	AMBC/PBPA, Run 5

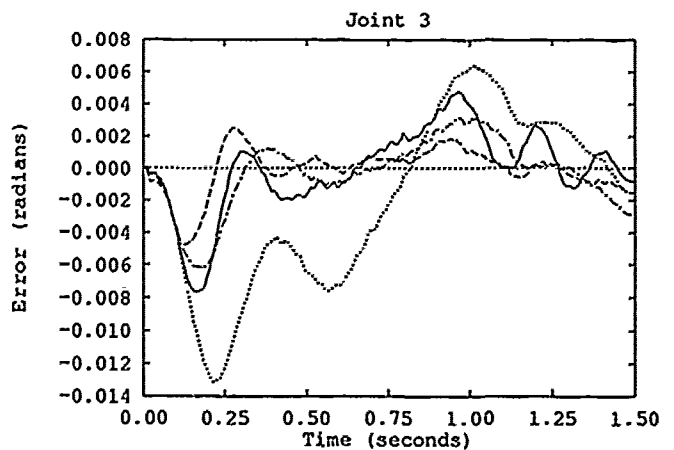
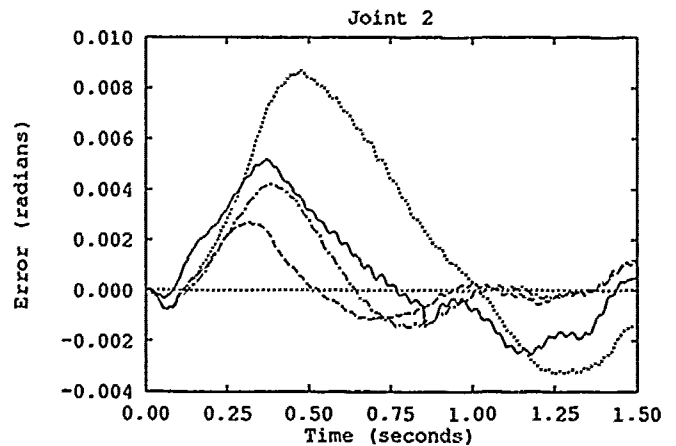
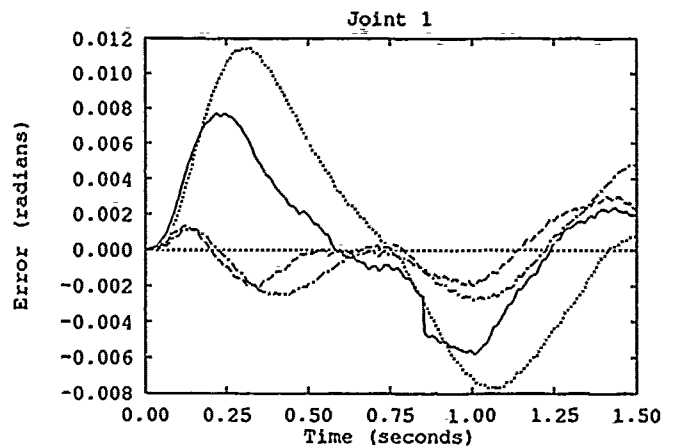


Figure A.38. Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 1, No Payload

—	Stiff PD Gains, Run 1
- - -	Stiff PD Gains, Run 5
...	Soft PD Gains, Run 1
- . - .	Soft PD Gains, Run 5

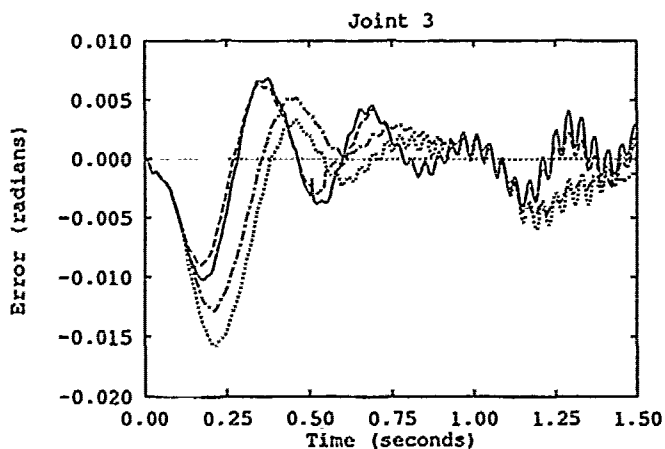
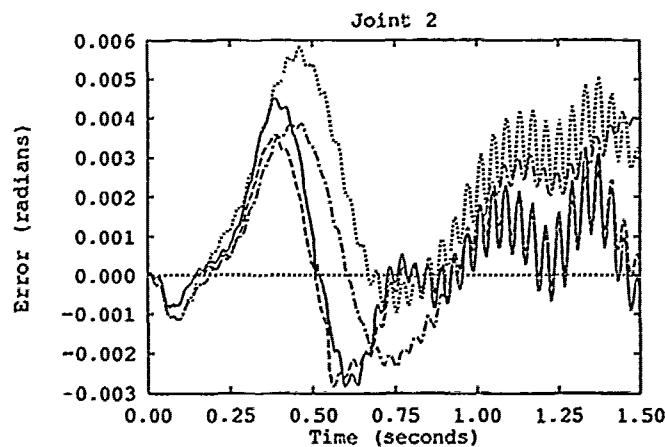
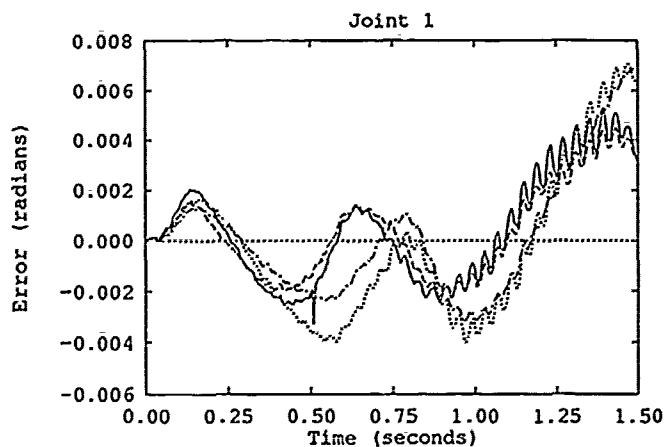


Figure A.39. Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 1, 2KG Payload

—	Stiff PD Gains, Run 1
---	Stiff PD Gains, Run 5
...	Soft PD Gains, Run 1
-.-.-	Soft PD Gains, Run 5

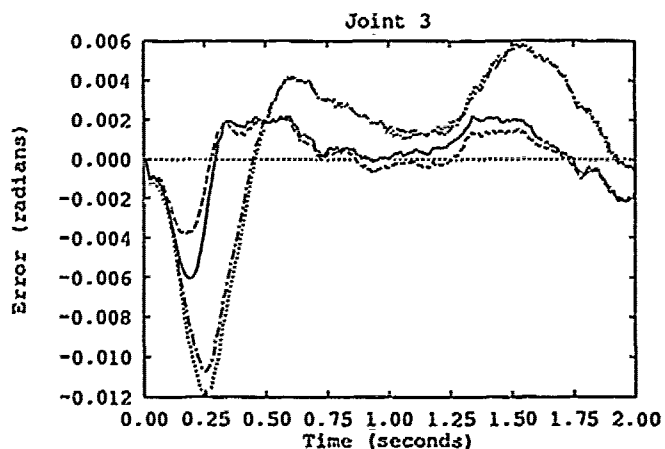
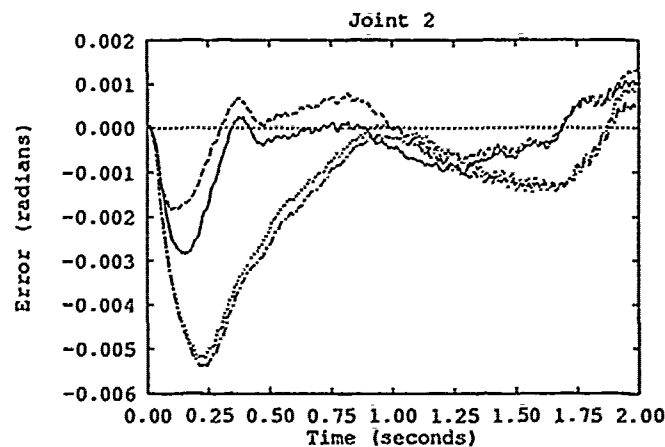
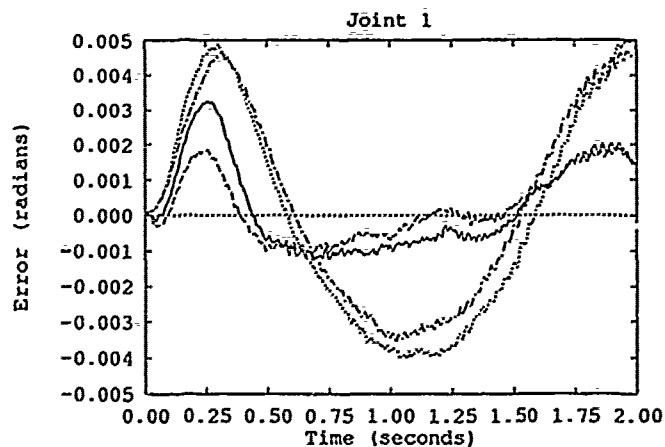


Figure A.40. Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 3, No Payload

—	Stiff PD Gains, Run 1
---	Stiff PD Gains, Run 5
...	Soft PD Gains, Run 1
-.-.-	Soft PD Gains, Run 5

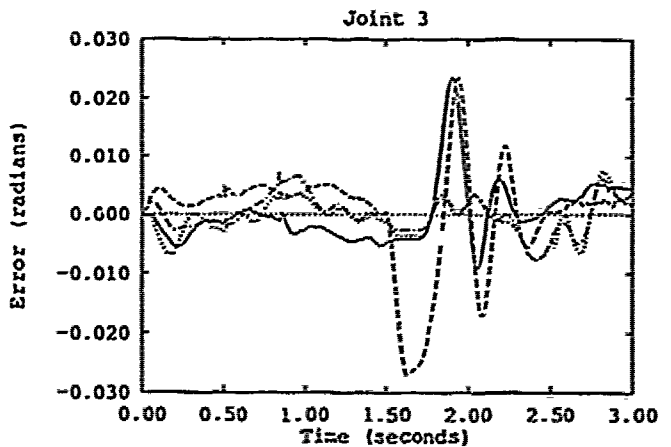
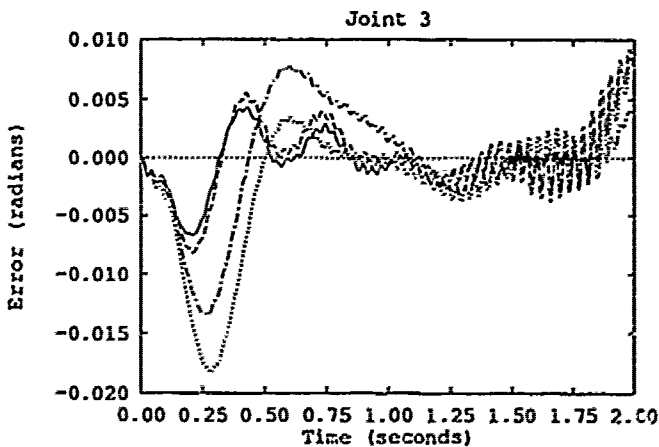
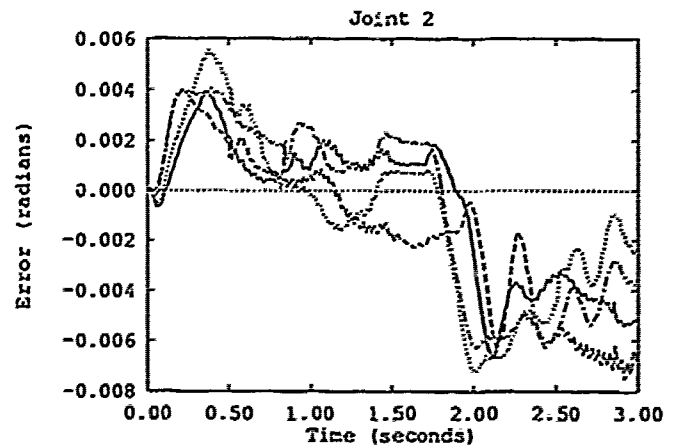
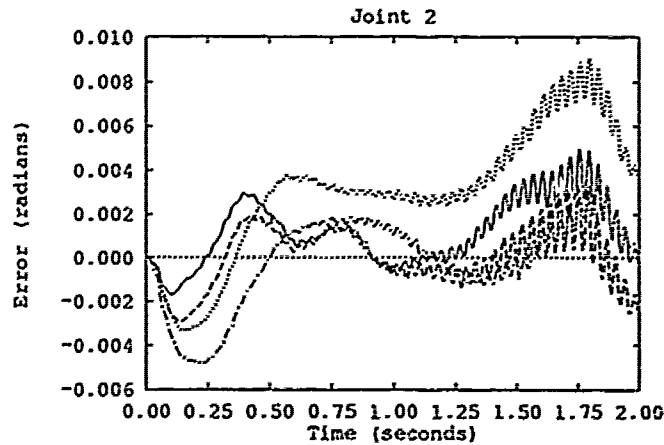
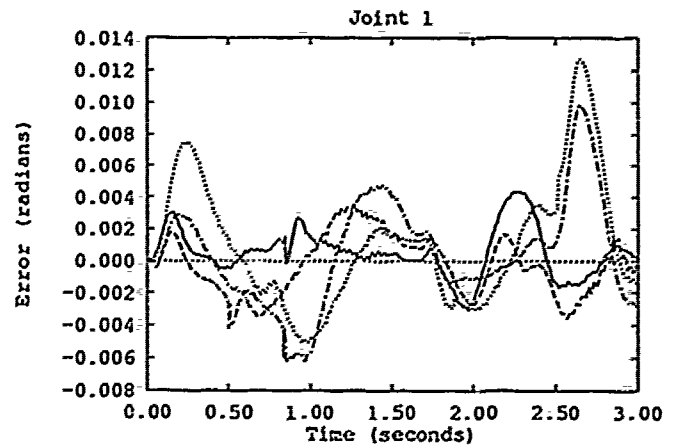
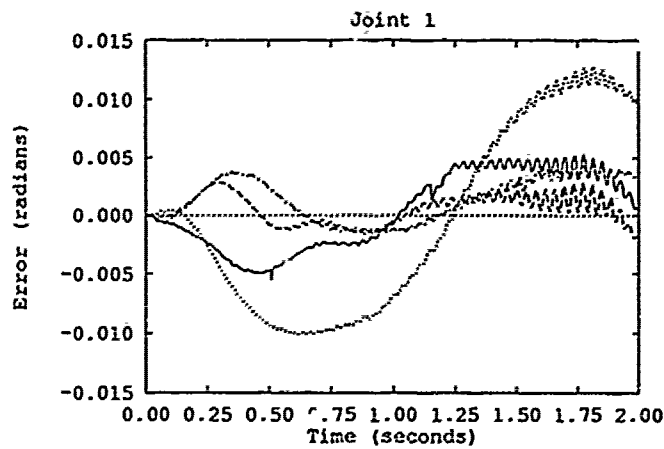


Figure A.41. Effect of Soft vs. Stiff PD Gains on the PBPA Algorithm, Trajectory 3, 2KG Payload

—	Stiff PD Gains, Run 1
- - -	Stiff PD Gains, Run 5
· · · · ·	Soft PD Gains, Run 1
- · - · -	Soft PD Gains, Run 5

Figure A.42. Two Way Trajectory Tracking Evaluation, No Payload, Trajectory 1

—	AMBC/H, Run 1
- - -	AMBC/H, Run 3
· · · · ·	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 3

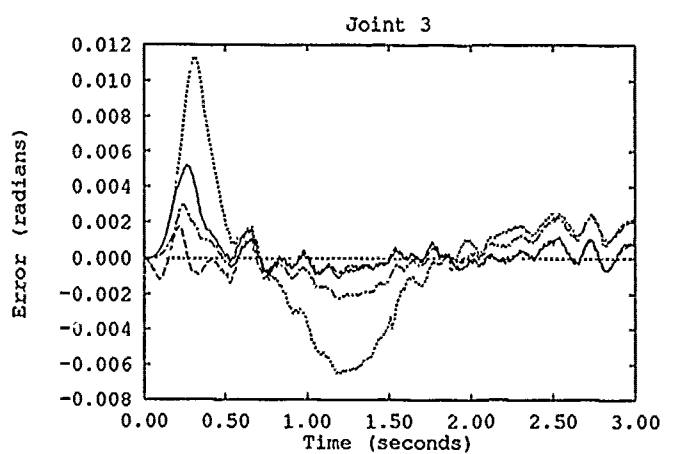
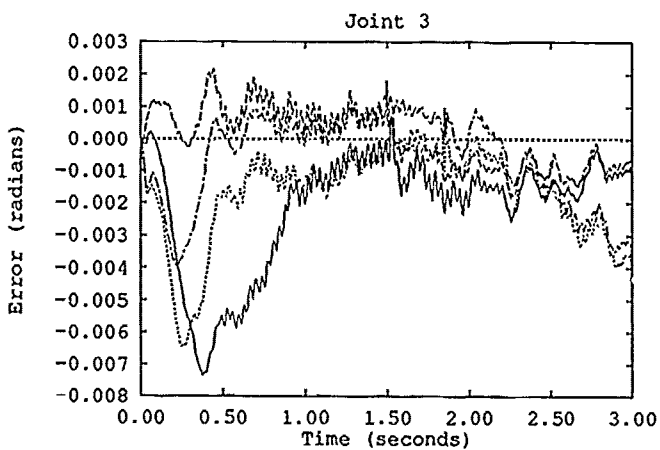
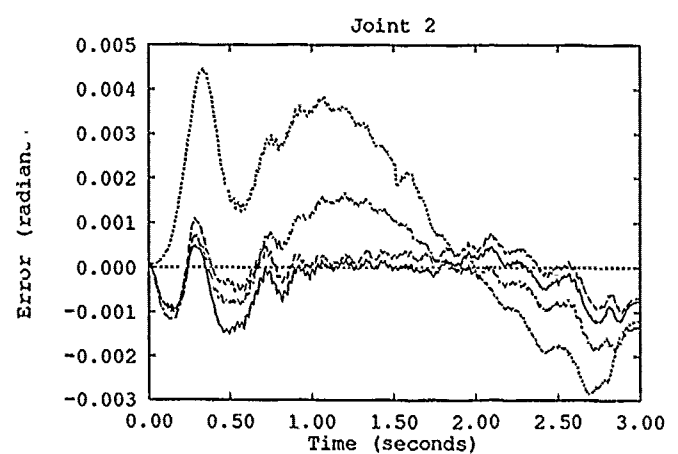
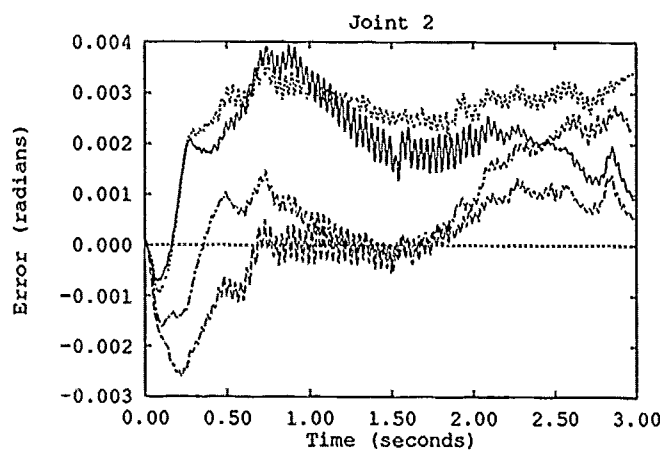
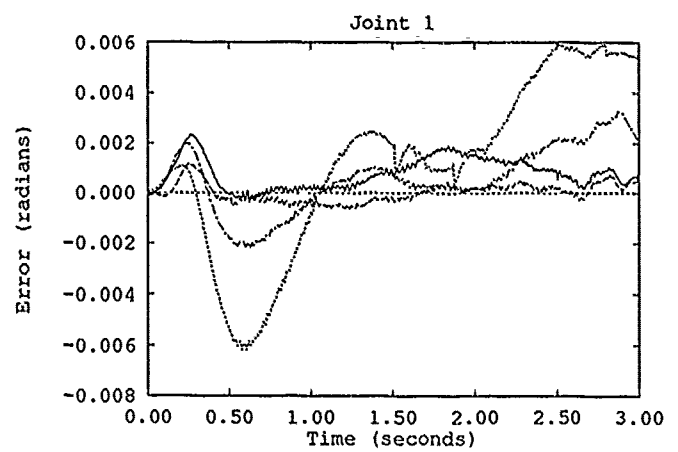
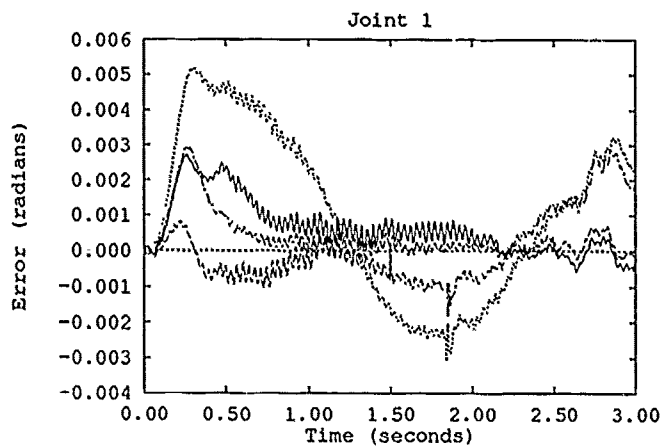


Figure A.43. Very Slow Tracking Evaluation, No Payload, Trajectory 1

Figure A.44. Very Slow Tracking Evaluation, No Payload, Trajectory 2

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
.....	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 5

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
.....	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 5

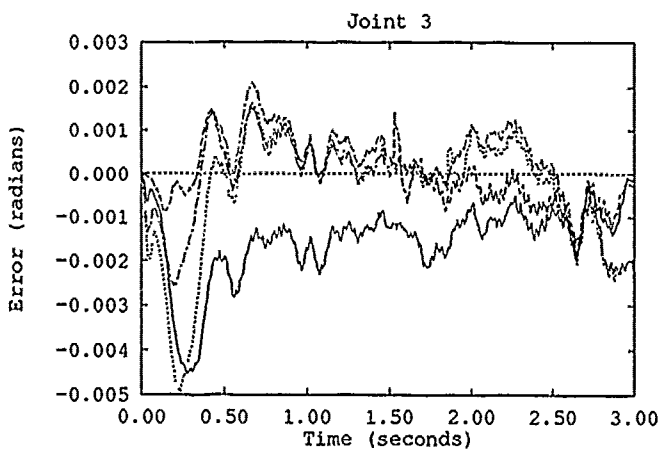
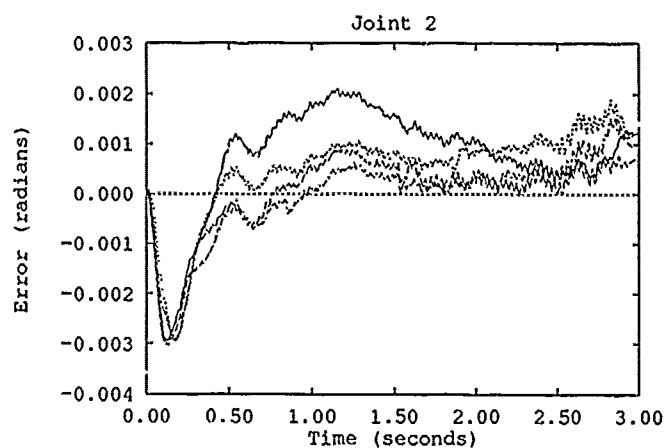
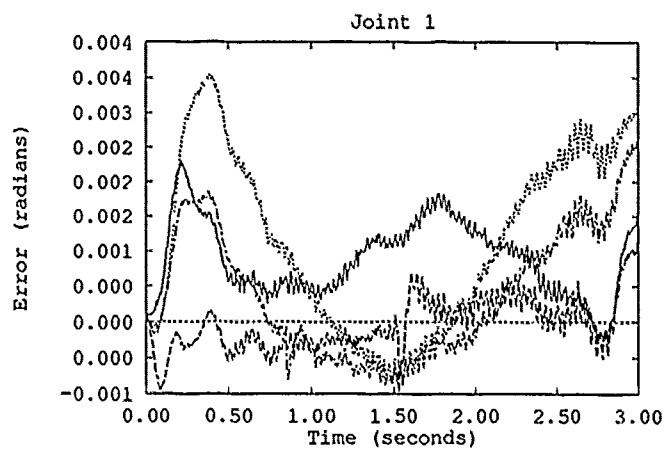


Figure A.45. Very Slow Tracking Evaluation, No Payload, Trajectory 3

—	AMBC/H, Run 1
- - -	AMBC/H, Run 5
...	AMBC/PBPA, Run 1
- · - · -	AMBC/PBPA, Run 5

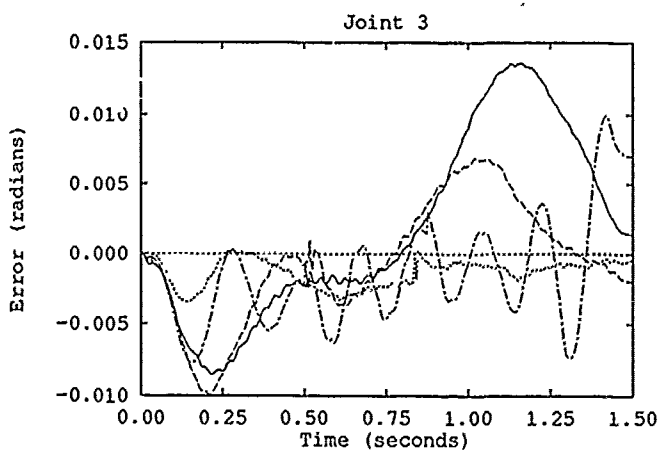
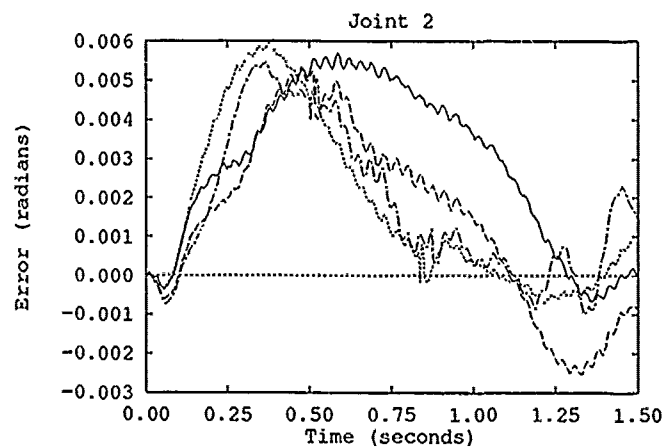
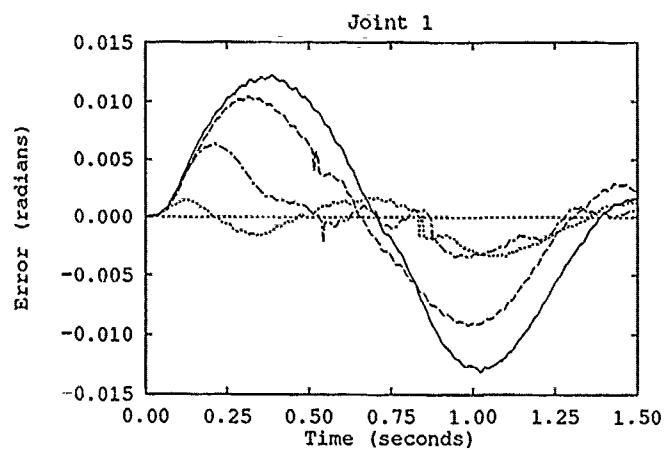


Figure A.46. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 1

—	Scale Factor = 0.001
- - -	Scale Factor = 0.005
...	Scale Factor = 0.01
- · - · -	Scale Factor = 0.03

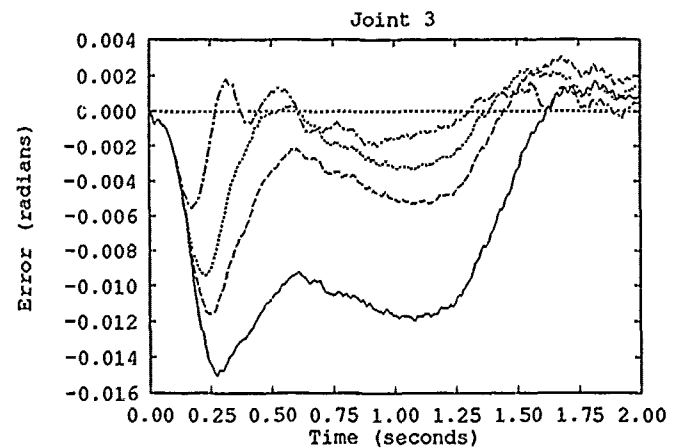
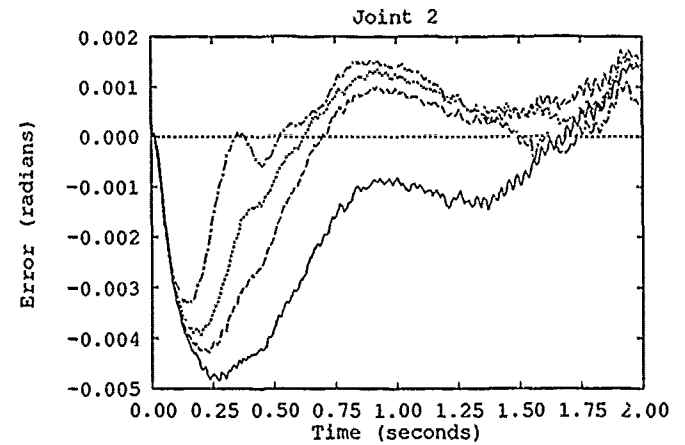
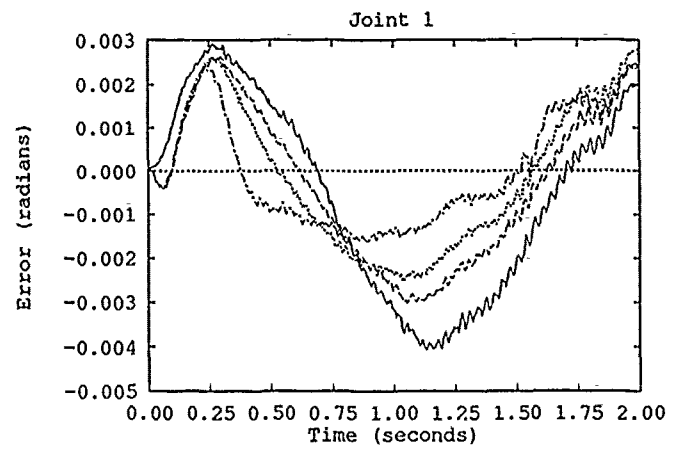
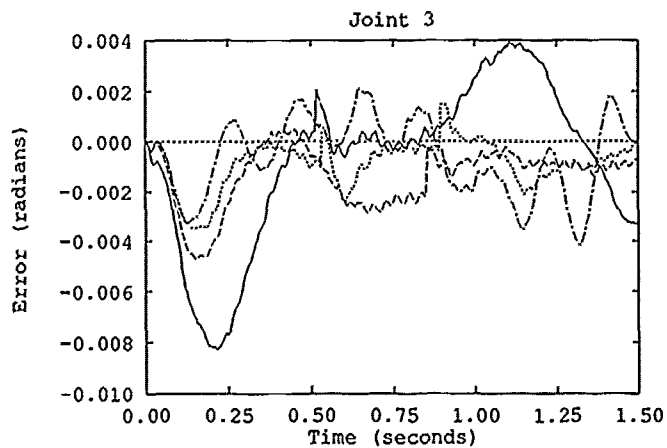
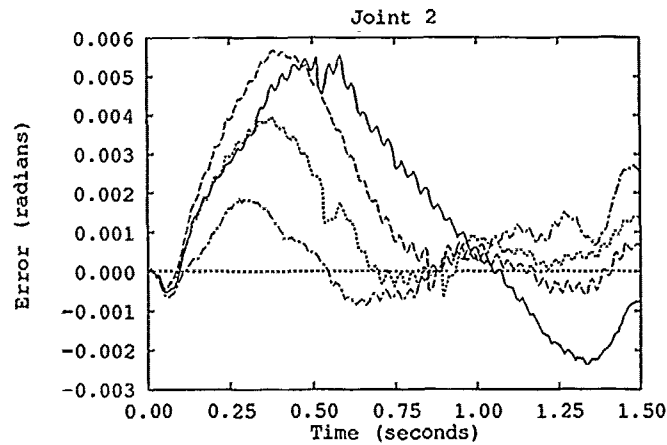
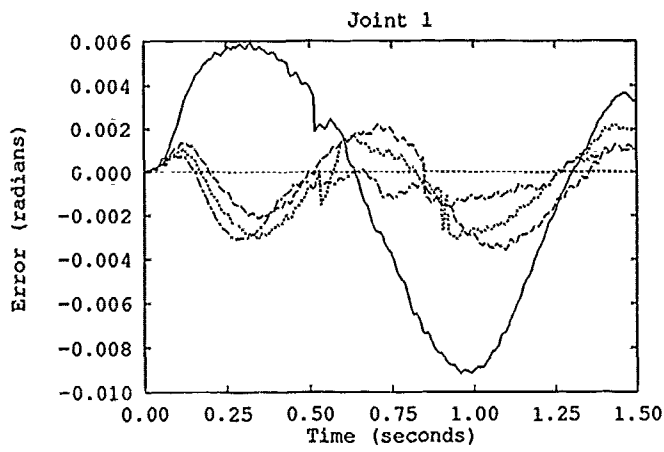


Figure A.47. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 1, Run 5

—	Scale Factor = 0.001
- - -	Scale Factor = 0.005
.....	Scale Factor = 0.01
- · - · -	Scale Factor = 0.03

Figure A.48. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 3, Run 1

—	Scale Factor = 0.001
- - -	Scale Factor = 0.005
.....	Scale Factor = 0.01
- · - · -	Scale Factor = 0.03

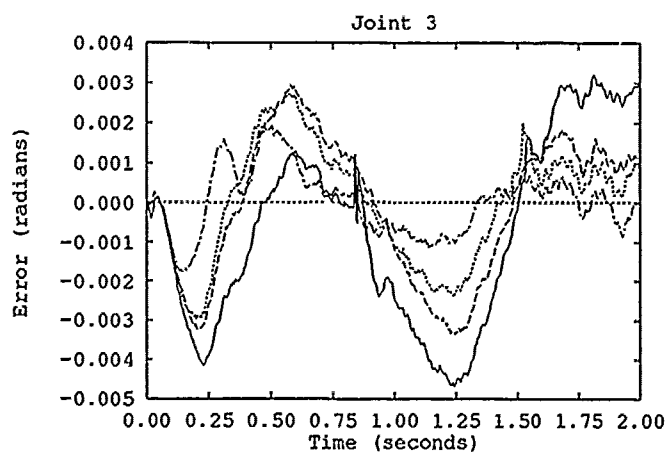
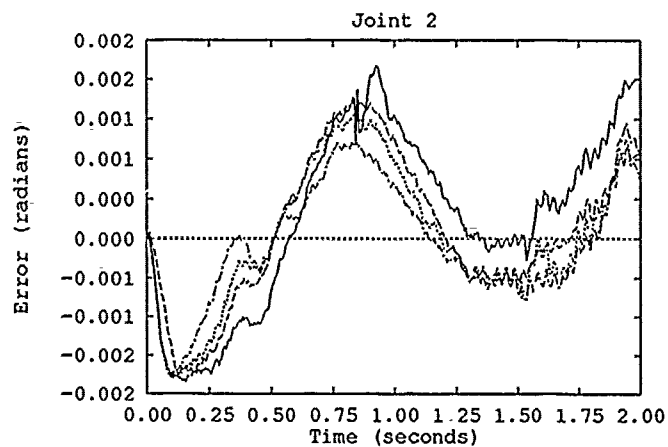
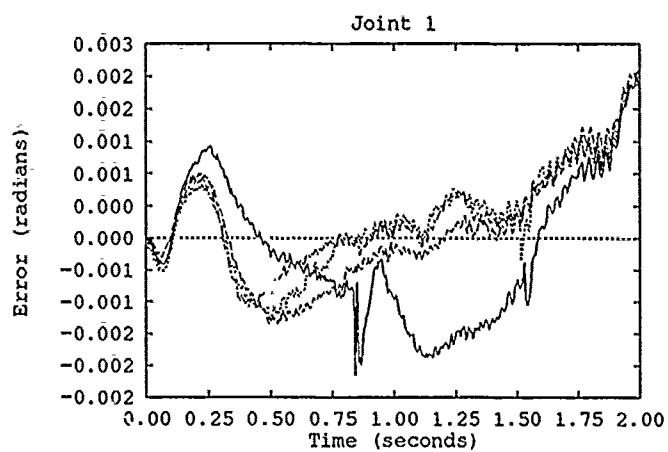


Figure A.49. Effect of Varying Γ^{-1} Scale Factor for AMBC/PBPA Algorithm, Trajectory 3, Run 5

—	Scale Factor = 0.001
----	Scale Factor = 0.005
.....	Scale Factor = 0.01
-.-.-.-	Scale Factor = 0.03

Appendix B. *PUMA-560, Three Degree of Freedom Values*

As discussed in section 3.2.2, the following are the actual values used for the regressor matrix, p^* and the k_i matrices in the AMBC/PBPA algorithm.

B.1 The Non-linear Function Vector, $n0$

```
n0(1) = QDD1
n0(2) = QDD1*sin2
n0(3) = QDD1*cos2**2
n0(4) = QDD1*cos2*cos23
n0(5) = QDD1*cos2*sin23
n0(6) = QDD1*cos23
n0(7) = QDD1*sin23
n0(8) = QDD1*cos23*sin23
n0(9) = QDD1*sin23**2
n0(10) = QDD2
n0(11) = QDD2*sin2
n0(12) = QDD2*cos3
n0(13) = QDD2*sin3
n0(14) = QDD2*cos23
n0(15) = QDD2*sin23
n0(16) = QDD3
n0(17) = QDD3*cos3
n0(18) = QDD3*sin3
n0(19) = QDD3*cos23
n0(20) = QDD3*sin23
n0(21) = QD1
n0(22) = QD1**2
n0(23) = QD1**2*cos2
n0(24) = QD1**2*cos2*sin2
n0(25) = QD1**2*cos2*cos23
n0(26) = QD1**2*cos2*sin23
n0(27) = QD1**2*sin2*sin23
n0(28) = QD1**2*sin2*sin3*sin23
n0(29) = QD1**2*cos23**2
n0(30) = QD1**2*sin3
n0(31) = QD1**2*cos3*sin3
```

```

n0(32) = QD1*QD2
n0(33) = QD1*QD2*cos2
n0(34) = QD1*QD2*cos2*sin2
n0(35) = QD1*QD2*cos2*cos23
n0(36) = QD1*QD2*cos2*sin23
n0(37) = QD1*QD2*sin2*sin23
n0(38) = QD1*QD2*sin2*sin3*sin23
n0(39) = QD1*QD2*cos23**2
n0(40) = QD1*QD2*sin3
n0(41) = QD1*QD2*cos3*sin3
n0(42) = QD1*QD3
n0(43) = QD1*QD3*cos2*sin2
n0(44) = QD1*QD3*cos2*cos23
n0(45) = QD1*QD3*cos2*sin23
n0(46) = QD1*QD3*sin2*sin3*sin23
n0(47) = QD1*QD3*cos23**2
n0(48) = QD1*QD3*cos3*sin3
n0(49) = QD2
n0(50) = QD2**2*cos3
n0(51) = QD2*QD3*cos3
n0(52) = QD2*QD3*sin3
n0(53) = QD2*QD3*cos23
n0(54) = QD2*QD3*sin23
n0(55) = QD2**2*cos2
n0(56) = QD2**2*cos23
n0(57) = QD2**2*sin23
n0(58) = QD3
n0(59) = QD3**2*cos3
n0(60) = QD3**2*sin3
n0(61) = QD3**2*cos23
n0(62) = QD3**2*sin23
n0(63) = cos2
n0(64) = sin2
n0(65) = cos23
n0(66) = sin23
n0(67) = 1
n0(68) = QD2**2*sin3

```

B.2 Constant Matrices of Kinematic Parameters, k_i

In the interest of space, zero values are not shown.

```

k1t[[1,44]] = .7766
k1t[[21,38]] = 1
k1t[[1,1]] = 1
k1t[[1,6]] = 1
k1t[[1,23]] = 1
k1t[[1,24]] = 1
k1t[[1,30]] = 1
1t[[67,41]] = 1
k1t[[1,18]] = 2
k1t[[1,26]] = 2
k1t[[32,20]] = -2
k1t[[42,20]] = -2
k1t[[33,3]] = -4
k1t[[55,4]] = 1
k1t[[3,5]] = 1
k1t[[3,6]] = -1
k1t[[3,7]] = 1
k1t[[3,16]] = 1
k1t[[3,2]] = 2
k1t[[56,22]] = 1
k1t[[53,22]] = 2
k1t[[61,22]] = 1
k1t[[56,25]] = 1
k1t[[53,25]] = 2
k1t[[61,25]] = 1
k1t[[56,19]] = 1
k1t[[53,19]] = 2
k1t[[61,19]] = 1
k1t[[14,27]] = -1
k1t[[19,27]] = -1
k1t[[4,17]] = 2
k1t[[4,13]] = 2
k1t[[35,15]] = 2
k1t[[44,15]] = 2
k1t[[39,20]] = 4
k1t[[47,20]] = 4
k1t[[11,12]] = 1
k1t[[11,14]] = 1
k1t[[11,4]] = 1
k1t[[34,5]] = -2
k1t[[34,6]] = 2

```

$k1t[[34,7]] = -2$
 $k1t[[34,16]] = -2$
 $k1t[[34,23]] = -2$
 $k1t[[43,23]] = -2$
 $k1t[[34,28]] = 2$
 $k1t[[43,28]] = 2$
 $k1t[[34,30]] = -2$
 $k1t[[43,30]] = -2$
 $k1t[[34,2]] = -4$
 $k1t[[34,18]] = -4$
 $k1t[[43,18]] = -4$
 $k1t[[40,17]] = 2$
 $k1t[[40,13]] = 2$
 $k1t[[41,23]] = -2$
 $k1t[[48,23]] = -2$
 $k1t[[41,28]] = 2$
 $k1t[[48,28]] = 2$
 $k1t[[41,30]] = -2$
 $k1t[[48,30]] = -2$
 $k1t[[41,18]] = -4$
 $k1t[[48,18]] = -4$
 $k1t[[15,22]] = 1$
 $k1t[[20,22]] = 1$
 $k1t[[15,25]] = 1$
 $k1t[[20,19]] = 1$
 $k1t[[57,27]] = 1$
 $k1t[[54,27]] = 2$
 $k1t[[62,27]] = 1$
 $k1t[[36,17]] = -4$
 $k1t[[45,17]] = -2$
 $k1t[[36,13]] = -4$
 $k1t[[45,13]] = -2$
 $k1t[[5,15]] = 2$
 $k1t[[8,20]] = 2$
 $k1t[[37,15]] = -2$
 $k1t[[38,23]] = 4$
 $k1t[[46,23]] = 4$
 $k1t[[38,28]] = -4$
 $k1t[[46,28]] = -4$
 $k1t[[38,30]] = 4$
 $k1t[[46,30]] = 4$
 $k1t[[38,18]] = 8$

$k1t[[46,18]] = 8$
 $k1t[[9,23]] = -1$
 $k1t[[9,28]] = 1$
 $k1t[[9,30]] = -1$
 $k1t[[9,18]] = -2$

$k2t[[10,44]] = 2.3616$
 $k2t[[4,39]] = 1$
 $k2t[[10,5]] = 1$
 $k2t[[10,8]] = 1$
 $k2t[[10,29]] = 1$
 $k2t[[16,29]] = 1$
 $k2t[[10,16]] = 1$
 $k2t[[10,23]] = 1$
 $k2t[[16,23]] = 1$
 $k2t[[67,42]] = 1$
 $k2t[[10,2]] = 2$
 $k2t[[10,18]] = 2$
 $k2t[[16,18]] = 2$
 $k2t[[22,20]] = 1$
 $k2t[[63,31]] = -1$
 $k2t[[63,34]] = -1$
 $k2t[[63,32]] = -1$
 $k2t[[23,3]] = 2$
 $k2t[[12,17]] = 2$
 $k2t[[17,17]] = 1$
 $k2t[[12,13]] = 2$
 $k2t[[17,13]] = 1$
 $k2t[[31,15]] = 2$
 $k2t[[39,15]] = 1$
 $k2t[[65,35]] = -1$
 $k2t[[65,36]] = -1$
 $k2t[[6,27]] = -1$
 $k2t[[25,15]] = -1$
 $k2t[[29,20]] = -2$
 $k2t[[2,12]] = 1$
 $k2t[[64,33]] = 1$
 $k2t[[2,14]] = 1$
 $k2t[[24,5]] = 1$
 $k2t[[24,6]] = -1$
 $k2t[[24,7]] = 1$

$k2t[[24,16]] = 1$
 $k2t[[24,23]] = 1$
 $k2t[[24,28]] = -1$
 $k2t[[24,30]] = 1$
 $k2t[[24,2]] = 2$
 $k2t[[24,18]] = 2$
 $k2t[[30,17]] = -1$
 $k2t[[52,17]] = -2$
 $k2t[[60,17]] = -1$
 $k2t[[30,13]] = -1$
 $k2t[[52,13]] = -2$
 $k2t[[60,13]] = -1$
 $k2t[[13,15]] = 2$
 $k2t[[18,15]] = 1$
 $k2t[[31,23]] = 1$
 $kt[[31,18]] = 2$
 $k2t[[7,22]] = 1$
 $k2t[[7,25]] = 1$
 $k2t[[7,19]] = 1$
 $k2t[[66,37]] = -1$
 $k2t[[26,17]] = 2$
 $k2t[[26,13]] = 2$
 $k2t[[27,15]] = 1$
 $k2t[[28,23]] = -2$
 $k2t[[28,28]] = 2$
 $k2t[[28,30]] = -2$
 $k2t[[28,18]] = -4$

$k3t[[16,44]] = .5827$
 $k3t[[58,40]] = 1$
 $k3t[[10,29]] = 1$
 $k3t[[16,29]] = 1$
 $k3t[[10,23]] = 1$
 $k3t[[10,18]] = 2$
 $k3t[[16,18]] = 2$
 $k3t[[22,20]] = 1$
 $k3t[[12,17]] = 1$
 $k3t[[12,13]] = 1$
 $k3t[[50,15]] = -1$
 $k3t[[65,35]] = -1$
 $k3t[[65,36]] = -1$

```

k3t[[6,27]] = -1
k3t[[25,15]] = -1
k3t[[29,20]] = -2
k3t[[24,23]] = 1
k3t[[24,28]] = -1
k3t[[24,30]] = 1
k3t[[24,18]] = 2
k3t[[68,17]] = 1
k3t[[68,13]] = 1
k3t[[13,15]] = 1
k3t[[31,23]] = 1
k3t[[31,28]] = -1
k3t[[31,30]] = 1
k3t[[31,18]] = 2
k3t[[7,22]] = 1
k3t[[7,25]] = 1
k3t[[7,19]] = 1
k3t[[66,37]] = -1
k3t[[26,17]] = 1
k3t[[26,13]] = 1
k3t[[28,23]] = -2
k3t[[28,28]] = 2

```

B.3 The Non-reduced Parameter Set, p

The physical values are as found in [27].

```

p(1)  = m1*kyy1
p(2)  = m2*aa2*xbar2
p(3)  = m2*aa2*ybar2
p(4)  = m2*aa2*zbar2
p(5)  = m2*aa2**2
p(6)  = m2*kxx2
p(7)  = m2*kyy2
p(8)  = m2*kzz2
p(9)  = m2*kxx2
p(10) = m3*xbar3
p(11) = m3*ybar3

```

$p(12) = m3*aa2*dd3$
 $p(13) = m3*aa2*xbar3$
 $p(14) = m3*aa2*ybar3$
 $p(15) = m3*aa2*zbar3$
 $p(16) = m3*aa2**2$
 $p(17) = m3*aa2*aa3$
 $p(18) = m3*aa3*xbar3$
 $p(19) = m3*aa3*ybar3$
 $p(20) = m3*aa3*zbar3$
 $p(21) = m3*aa3$
 $p(22) = m3*aa3*dd3$
 $p(23) = m3*aa3**2$
 $p(24) = m3*dd3**2$
 $p(25) = m3*dd3*xbar3$
 $p(26) = m3*dd3*ybar3$
 $p(27) = m3*dd3*zbar3$
 $p(28) = m3*kxx3$
 $p(29) = m3*kyy3$
 $p(30) = m3*kzz3$
 $p(31) = m2*GRAV*aa2$
 $p(32) = m2*GRAV*xbar2$
 $p(33) = m2*GRAV*ybar2$
 $p(34) = m3*GRAV*aa2$
 $p(35) = m3*GRAV*aa3$
 $p(36) = m3*GRAV*xbar3$
 $p(37) = m3*GRAV*zbar3$
 $p(38) = bn21$
 $p(39) = bn22$
 $p(40) = bn23$
 $p(41) = sftor1$
 $p(42) = sftor2$
 $p(43) = sftor3$
 $p(44) = 1$

B.4 The AMBC/PBPA Regressor Matrix

$Y(1,1) = 0. - 1.291916379341979*QD1*QD2*cos2 +$
 $1.223543615235044*10**-16*QDD2*sin2 -$
 $6.95748284379206*10**-17*(-(QD2**2*cos2) - QDD2*sin2)$

$$Y(1,2) = 0. - 3.785624396158174*QD1*QD2*cos2$$

$$Y(1,3) = 0.$$

$$Y(1,4) = 0.$$

$$Y(1,5) = 1.0$$

$$Y(1,6) = QD1$$

$$Y(1,7) = 0.$$

$$Y(1,8) = 0.$$

$$Y(1,9) = 0.$$

$$Y(1,10) = 0. - 1.434315310235834*10**-16*QD1*QD2*cos2 + \\ 1.102070113928819*QDD2*sin2 - \\ 0.6266743428547707*(-(QD2**2*cos2) - QDD2*sin2)$$

$$Y(1,11) = 0. - 0.836251354856464*QDD2*sin2 - \\ 0.7792812509020992*(-(QD2**2*cos2) - QDD2*sin2)$$

$$Y(1,12) = -0.295459647105532*QDD1 + \\ 0.01225149578425218*(-QDD1 + 2*QDD1*cos2**2 - \\ 4*QD1*QD2*cos2*sin2) - \\ 1.377020985552159*(-QDD1 + QDD1*cos2**2 - \\ 2*QD1*QD2*cos2*sin2) + \\ 3.939813994754334*(-2*QD1*QD2*cos2*sin2 - \\ 2*QD1*QD3*cos2*sin2 - 2*QD1*QD2*cos3*sin3 - \\ 2*QD1*QD3*cos3*sin3 + 4*QD1*QD2*sin2*sin3*sin23 + \\ 4*QD1*QD3*sin2*sin3*sin23 - QDD1*sin23**2) + \\ 1.613375882778644*(2*QD1*QD2*cos2*sin2 + \\ 2*QD1*QD3*cos2*sin2 + \\ 2*QD1*QD2*cos3*sin3 + 2*QD1*QD3*cos3*sin3 - \\ 4*QD1*QD2*sin2*sin3*sin23 - \\ 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) + \\ 3.636549415076395*(-QDD1 + 2*QD1*QD2*cos2*sin2 + \\ 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\ 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\ 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2)$$

$$Y(1,13) = 0.$$

$$\begin{aligned} Y(1,14) = & 0.03071435139678527*QDD1 + \\ & 0.0971847547297091*(-QDD1 + 2*QDD1*cos2**2 - \\ & 4*QD1*QD2*cos2*sin2) + \\ & 0.907177973662084*(-QDD1 + QDD1*cos2**2 - \\ & 2*QD1*QD2*cos2*sin2) - \\ & 0.944620431342006*(-2*QD1*QD2*cos2*sin2 - \\ & 2*QD1*QD3*cos2*sin2 - 2*QD1*QD2*cos3*sin3 - \\ & 2*QD1*QD3*cos3*sin3 + 4*QD1*QD2*sin2*sin3*sin23 + \\ & 4*QD1*QD3*sin2*sin3*sin23 - QDD1*sin23**2) + \\ & 1.656684419083955*(2*QD1*QD2*cos2*sin2 + \\ & 2*QD1*QD3*cos2*sin2 + \\ & 2*QD1*QD2*cos3*sin3 + 2*QD1*QD3*cos3*sin3 - \\ & 4*QD1*QD2*sin2*sin3*sin23 - \\ & 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) - \\ & 1.105183688973447*(-QDD1 + 2*QD1*QD2*cos2*sin2 + \\ & 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\ & 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\ & 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) \end{aligned}$$

$$\begin{aligned} Y(1,15) = & -0.1907450736312253*QDD1 + \\ & 0.07015604633145501*(-QDD1 + 2*QDD1*cos2**2 - \\ & 4*QD1*QD2*cos2*sin2) + \\ & 0.58513755877427*(-QDD1 + QDD1*cos2**2 - \\ & 2*QD1*QD2*cos2*sin2) + \\ & 0.0950052880231652*(-2*QD1*QD2*cos2*sin2 - \\ & 2*QD1*QD3*cos2*sin2 - 2*QD1*QD2*cos3*sin3 - \\ & 2*QD1*QD3*cos3*sin3 + 4*QD1*QD2*sin2*sin3*sin23 + \\ & 4*QD1*QD3*sin2*sin3*sin23 - QDD1*sin23**2) - \\ & 0.001127205963008193*(2*QD1*QD2*cos2*sin2 + \\ & 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\ & 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\ & 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) - \\ & 1.476001365402235*(-QDD1 + 2*QD1*QD2*cos2*sin2 + \\ & 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\ & 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\ & 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) \end{aligned}$$

$$\begin{aligned} Y(1,16) = & -0.3680403457911835*QDD1 + \\ & 0.1524306276280933*(-QDD1 + 2*QDD1*cos2**2 - \end{aligned}$$

$$\begin{aligned}
& 4*QD1*QD2*cos2*sin2) - \\
& 1.367130671348102*(-QDD1 + QDD1*cos2**2 - \\
& 2*QD1*QD2*cos2*sin2) - \\
& 1.220605484649777*(-2*QD1*QD2*cos2*sin2 - \\
& 2*QD1*QD3*cos2*sin2 - 2*QD1*QD2*cos3*sin3 - \\
& 2*QD1*QD3*cos3*sin3 + 4*QD1*QD2*sin2*sin3*sin23 + \\
& 4*QD1*QD3*sin2*sin3*sin23 - QDD1*sin23**2) - \\
& 0.7599456368395224*(2*QD1*QD2*cos2*sin2 + \\
& 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\
& 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\
& 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) - \\
& 1.124898280211656*(-QDD1 + 2*QD1*QD2*cos2*sin2 + \\
& 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\
& 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\
& 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2)
\end{aligned}$$

$$\begin{aligned}
Y(1,17) = & 0. + 1.*(-2*QD1*QD2*cos2*cos23 - \\
& 2*QD1*QD3*cos2*cos23 - 2*QDD1*cos2*sin23 + \\
& 2*QD1*QD2*sin2*sin23)
\end{aligned}$$

$$\begin{aligned}
Y(1,18) = & -0.4217098280971802*QDD1 + \\
& 0.4935339141969876*(-QDD1 + 2*QDD1*cos2**2 - \\
& 4*QD1*QD2*cos2*sin2) + \\
& 0.820526775345378*(-QDD1 + QDD1*cos2**2 - \\
& 2*QD1*QD2*cos2*sin2) - \\
& 0.6732582230870278*(-2*QD1*QD2*cos2*sin2 - \\
& 2*QD1*QD3*cos2*sin2 - 2*QD1*QD2*cos3*sin3 - \\
& 2*QD1*QD3*cos3*sin3 + 4*QD1*QD2*sin2*sin3*sin23 + \\
& 4*QD1*QD3*sin2*sin3*sin23 - QDD1*sin23**2) - \\
& 0.2580049694968252*(2*QD1*QD2*cos2*sin2 + \\
& 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\
& 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\
& 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2) - \\
& 0.0893179992172112*(-QDD1 + 2*QD1*QD2*cos2*sin2 + \\
& 2*QD1*QD3*cos2*sin2 + 2*QD1*QD2*cos3*sin3 + \\
& 2*QD1*QD3*cos3*sin3 - 4*QD1*QD2*sin2*sin3*sin23 - \\
& 4*QD1*QD3*sin2*sin3*sin23 + QDD1*sin23**2)
\end{aligned}$$

$$Y(1,19) = 0.$$

$$\begin{aligned}
Y(1,20) = & 0. - 1.*(QDD2*cos23 + QDD3*cos23 - QD2**2*sin23 - \\
& 2*QD2*QD3*sin23 - QD3**2*sin23)
\end{aligned}$$

$$\begin{aligned}
Y(1,21) = & -0.4064372364733833*QDD1 - \\
& 0.6845964743997872*(-QDD1 + 2*QDD1*\cos2**2 - \\
& 4*QD1*QD2*\cos2*\sin2) + \\
& 1.181585758700407*(-QDD1 + QDD1*\cos2**2 - \\
& 2*QD1*QD2*\cos2*\sin2) - \\
& 1.176170709788934*(-2*QD1*QD2*\cos2*\sin2 - \\
& 2*QD1*QD3*\cos2*\sin2 - 2*QD1*QD2*\cos3*\sin3 - \\
& 2*QD1*QD3*\cos3*\sin3 + 4*QD1*QD2*\sin2*\sin3*\sin23 + \\
& 4*QD1*QD3*\sin2*\sin3*\sin23 - QDD1*\sin23**2) - \\
& 0.0912676865211562*(2*QD1*QD2*\cos2*\sin2 + \\
& 2*QD1*QD3*\cos2*\sin2 + 2*QD1*QD2*\cos3*\sin3 + \\
& 2*QD1*QD3*\cos3*\sin3 - 4*QD1*QD2*\sin2*\sin3*\sin23 - \\
& 4*QD1*QD3*\sin2*\sin3*\sin23 + QDD1*\sin23**2) - \\
& 0.923105350898833*(-QDD1 + 2*QD1*QD2*\cos2*\sin2 + \\
& 2*QD1*QD3*\cos2*\sin2 + 2*QD1*QD2*\cos3*\sin3 + \\
& 2*QD1*QD3*\cos3*\sin3 - 4*QD1*QD2*\sin2*\sin3*\sin23 - \\
& 4*QD1*QD3*\sin2*\sin3*\sin23 + QDD1*\sin23**2)
\end{aligned}$$

$$\begin{aligned}
Y(1,22) = & 0. - 1.*(2*QD1*QD2 + 2*QD1*QD3 - \\
& 4*QD1*QD2*\cos23**2 - \\
& 4*QD1*QD3*\cos23**2 - 2*QDD1*\cos23*\sin23)
\end{aligned}$$

$$\begin{aligned}
Y(1,23) = & 0. - 1.414213562373096*(-2*QDD1*\cos2*\cos23 - \\
& 2*QD1*QD2*\sin3 + 4*QD1*QD2*\cos2*\sin23 + \\
& 2*QD1*QD3*\cos2*\sin23)
\end{aligned}$$

$$\begin{aligned}
Y(1,24) = & 0. - 1.732050807568877*(-(QD2**2*\cos23) - \\
& 2*QD2*QD3*\cos23 - QD3**2*\cos23 - QDD2*\sin23 - \\
& QDD3*\sin23)
\end{aligned}$$

$$\begin{aligned}
Y(2,1) = & 0. - 1.639223448129551*\cos2 + \\
& 0.6459581896709893*QD1**2*\cos2
\end{aligned}$$

$$\begin{aligned}
Y(2,2) = & 0. + 0.5594162020376836*\cos2 + \\
& 1.892812198079087*QD1**2*\cos2
\end{aligned}$$

$$Y(2,3) = 0. - 1.*\sin23$$

$$Y(2,4) = 1.0$$

$$Y(2,5) = 0.$$

$$Y(2,6) = 0.$$

$$Y(2,7) = 0.$$

$$Y(2,8) = QD2$$

$$Y(2,9) = 0.$$

$$Y(2,10) = 0. + 1.72874445678359*QDD1*\sin2$$

$$Y(2,11) = 0. - 0.1069701039543647*QDD1*\sin2$$

$$\begin{aligned} Y(2,12) = & -0.898477340464112*QDD2 + \\ & 3.484890831498359*(-QDD2 - QDD3) + \\ & 0.90984632651194*(-QDD2 - 2*QD1**2*\cos2*\sin2) - \\ & 0.4671746590402205*(QDD2 - QD1**2*\cos2*\sin2) + \\ & 5.053141200540136*(-(QD1**2*\cos2*\sin2) - \\ & QD1**2*\cos3*\sin3 + \\ & 2*QD1**2*\sin2*\sin3*\sin23) - \\ & 1.871514948719715*(-QDD2 - QDD3 - 2*QD1**2*\cos2*\sin2 - \\ & 2*QD1**2*\cos3*\sin3 + 4*QD1**2*\sin2*\sin3*\sin23) \end{aligned}$$

$$Y(2,13) = 0. - 1.*\sin2$$

$$\begin{aligned} Y(2,14) = & 0.0934007368769639*QDD2 - \\ & 0.6098328552114092*(-QDD2 - QDD3) - \\ & 0.6695751522611962*(-QDD2 - 2*QD1**2*\cos2*\sin2) + \\ & 0.2376028214008885*(QDD2 - QD1**2*\cos2*\sin2) - \\ & 3.036913387138213*(-(QD1**2*\cos2*\sin2) - \\ & QD1**2*\cos3*\sin3 + \\ & 2*QD1**2*\sin2*\sin3*\sin23) + \\ & 2.266517274295362*(-QDD2 - QDD3 - 2*QD1**2*\cos2*\sin2 - \\ & 2*QD1**2*\cos3*\sin3 + 4*QD1**2*\sin2*\sin3*\sin23) \end{aligned}$$

$$\begin{aligned} Y(2,15) = & -0.580045796919265*QDD2 + \\ & 0.3134364897250736*(-QDD2 - QDD3) - \\ & 0.4368624034038167*(-QDD2 - 2*QD1**2*\cos2*\sin2) + \\ & 0.148275155370453*(QDD2 - QD1**2*\cos2*\sin2) - \\ & 0.943006468012244*(-(QD1**2*\cos2*\sin2) - \end{aligned}$$

$$\begin{aligned}
& QD1**2*cos3*sin3 + \\
& 2*QD1**2*sin2*sin3*sin23) - \\
& 0.314563695688082*(-QDD2 - QDD3 - 2*QD1**2*cos2*sin2 - \\
& 2*QD1**2*cos3*sin3 + 4*QD1**2*sin2*sin3*sin23)
\end{aligned}$$

$$\begin{aligned}
Y(2,16) = & -1.119191450708806*QDD2 - \\
& 1.625505499223111*(-QDD2 - QDD3) + \\
& 0.809800029146671*(-QDD2 - 2*QD1**2*cos2*sin2) - \\
& 0.5573306422014298*(QDD2 - QD1**2*cos2*sin2) - \\
& 2.395358157168579*(-(QD1**2*cos2*sin2) - \\
& QD1**2*cos3*sin3 + \\
& 2*QD1**2*sin2*sin3*sin23) + \\
& 0.865559862383588*(-QDD2 - QDD3 - 2*QD1**2*cos2*sin2 - \\
& 2*QD1**2*cos3*sin3 + 4*QD1**2*sin2*sin3*sin23)
\end{aligned}$$

$$\begin{aligned}
Y(2,17) = & 0. + 1.*(-2*QD2*QD3*cos3 - QD3**2*cos3 + \\
& QD1**2*cos2*cos23 - 2*QDD2*sin3 - QDD3*sin3 - \\
& QD1**2*sin2*sin23)
\end{aligned}$$

$$\begin{aligned}
Y(2,18) = & -1.282397540605588*QDD2 - \\
& 0.235237993239572*(-QDD2 - QDD3) - \\
& 0.87604045969491*(-QDD2 - 2*QD1**2*cos2*sin2) - \\
& 0.0555136843495328*(QDD2 - QD1**2*cos2*sin2) + \\
& 0.3714692068874966*(-(QD1**2*cos2*sin2) - \\
& QD1**2*cos3*sin3 + \\
& 2*QD1**2*sin2*sin3*sin23) - \\
& 0.02276697625725266*(-QDD2 - QDD3 - \\
& 2*QD1**2*cos2*sin2 - \\
& 2*QD1**2*cos3*sin3 + 4*QD1**2*sin2*sin3*sin23)
\end{aligned}$$

$$Y(2,19) = 0. - 1.414213562373095*cos23$$

$$Y(2,20) = 0. - 1.*QDD1*cos23$$

$$\begin{aligned}
Y(2,21) = & -1.235954387915968*QDD2 - \\
& 1.010507615022058*(-QDD2 - QDD3) - \\
& 0.3313261895337464*(-QDD2 - 2*QD1**2*cos2*sin2) + \\
& 0.850259569166661*(QDD2 - QD1**2*cos2*sin2) - \\
& 1.676682184632862*(-(QD1**2*cos2*sin2) - \\
& QD1**2*cos3*sin3 + \\
& 2*QD1**2*sin2*sin3*sin23) + \\
& 0.919239928500904*(-QDD2 - QDD3 - 2*QD1**2*cos2*sin2 -
\end{aligned}$$

$$2*QD1**2*cos3*sin3 + 4*QD1**2*sin2*sin3*sin23)$$

$$Y(2,22) = 0. - 1.*(-QD1**2 + 2*QD1**2*cos23**2)$$

$$Y(2,23) = 0. - 1.414213562373096*(-2*QDD2*cos3 - QDD3*cos3 + QD1**2*sin3 + 2*QD2*QD3*sin3 + QD3**2*sin3 - 2*QD1**2*cos2*sin23)$$

$$Y(2,24) = 0. + 1.732050807568877*QDD1*sin23$$

$$Y(3,1) = 0.$$

$$Y(3,2) = 0.$$

$$Y(3,3) = 0. - 1.*sin23$$

$$Y(3,4) = 0.$$

$$Y(3,5) = 0.$$

$$Y(3,6) = 0.$$

$$Y(3,7) = 1.0$$

$$Y(3,8) = 0.$$

$$Y(3,9) = QD3$$

$$Y(3,10) = 0.$$

$$Y(3,11) = 0.$$

$$Y(3,12) = -1.571266692953117*(QDD2 + QDD3) - 0.02105459491276347*(2*QDD2 + 2*QDD3 + 2*QD1**2*cos2*sin2 + 2*QD1**2*cos3*sin3 - 4*QD1**2*sin2*sin3*sin23) + 1.648454916655984*(-(QD1**2*cos2*sin2) - QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23) - 0.3804528033808055*(0.5827*QDD3 - QD1**2*cos2*sin2 -$$

$$QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23)$$

$$Y(3,13) = 0.$$

$$\begin{aligned} Y(3,14) = & 1.226745384938757*(QDD2 + QDD3) - \\ & 1.441714902011356*(2*QDD2 + 2*QDD3 + \\ & 2*QD1**2*cos2*sin2 + \\ & 2*QD1**2*cos3*sin3 - 4*QD1**2*sin2*sin3*sin23) - \\ & 1.426858412589238*(-(QD1**2*cos2*sin2) - \\ & QD1**2*cos3*sin3 + \\ & 2*QD1**2*sin2*sin3*sin23) + \\ & 0.03954977001903942*(0.5827*QDD3 - QD1**2*cos2*sin2 - \\ & QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23) \end{aligned}$$

$$\begin{aligned} Y(3,15) = & -1.371048421615176*(QDD2 + QDD3) + \\ & 0.6860878137890928*(2*QDD2 + 2*QDD3 + \\ & 2*QD1**2*cos2*sin2 + \\ & 2*QD1**2*cos3*sin3 - 4*QD1**2*sin2*sin3*sin23) + \\ & 0.04565736647876157*(-(QD1**2*cos2*sin2) - \\ & QD1**2*cos3*sin3 + \\ & 2*QD1**2*sin2*sin3*sin23) - \\ & 0.245615598288984*(0.5827*QDD3 - QD1**2*cos2*sin2 - \\ & QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23) \end{aligned}$$

$$\begin{aligned} Y(3,16) = & 1.523064932932402*(QDD2 + QDD3) - \\ & 0.3815596480464399*(2*QDD2 + 2*QDD3 + \\ & 2*QD1**2*cos2*sin2 + \\ & 2*QD1**2*cos3*sin3 - 4*QD1**2*sin2*sin3*sin23) - \\ & 0.953445359461082*(-(QD1**2*cos2*sin2) - \\ & QD1**2*cos3*sin3 + \\ & 2*QD1**2*sin2*sin3*sin23) - \\ & 0.4739123690332005*(0.5827*QDD3 - QD1**2*cos2*sin2 - \\ & QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23) \end{aligned}$$

$$Y(3,17) = 0. + 1.*(QD2**2*cos3 + QD1**2*cos2*cos23 - QDD2*sin3)$$

$$\begin{aligned} Y(3,18) = & 0.957161576774482*(QDD2 + QDD3) - \\ & 0.349578303638828*(2*QDD2 + 2*QDD3 + \\ & 2*QD1**2*cos2*sin2 + \\ & 2*QD1**2*cos3*sin3 - 4*QD1**2*sin2*sin3*sin23) + \\ & 0.1697992859019009*(-(QD1**2*cos2*sin2) - \end{aligned}$$

$$\begin{aligned}
& QD1**2*cos3*sin3 + \\
& 2*QD1**2*sin2*sin3*sin23) - \\
& 0.5430206388065669*(0.5827*QDD3 - QD1**2*cos2*sin2 - \\
& QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23)
\end{aligned}$$

$$Y(3,19) = 0. - 1.414213562373095*cos23$$

$$Y(3,20) = 0. - 1.*QDD1*cos23$$

$$\begin{aligned}
Y(3,21) = & 1.034677675294261*(QDD2 + QDD3) - \\
& 0.471704994386553*(2*QDD2 + 2*QDD3 + \\
& 2*QD1**2*cos2*sin2 + \\
& 2*QD1**2*cos3*sin3 - 4*QD1**2*sin2*sin3*sin23) - \\
& 0.2582576467242971*(-(QD1**2*cos2*sin2) - \\
& QD1**2*cos3*sin3 + \\
& 2*QD1**2*sin2*sin3*sin23) - \\
& 0.5233546696798646*(0.5827*QDD3 - \\
& QD1**2*cos2*sin2 - \\
& QD1**2*cos3*sin3 + 2*QD1**2*sin2*sin3*sin23)
\end{aligned}$$

$$Y(3,22) = 0. - 1.*(-QD1**2 + 2*QD1**2*cos23**2)$$

$$\begin{aligned}
Y(3,23) = & 0. - 1.414213562373096*(-(QDD2*cos3) - \\
& QD2**2*sin3 - \\
& QD1**2*cos2*sin23)
\end{aligned}$$

$$Y(3,24) = 0. + 1.732050807568877*QDD1*sin23$$

B.5 The Principal Base Parameter Set, P^*

The physical values are as found in [13].

$$\begin{aligned}
\$P^*\$(1) = & 0.5464078160431836*GRAV*AA(2)*M(2) + \\
& 0.5464078160431836*GRAV*AA(2)*M(3) + \\
& 0.5464078160431836*GRAV*M(2)*XBAR(2) +
\end{aligned}$$

$$0.3229790948354945*AA(2)*M(2)*YBAR(2)$$

$$\begin{aligned} \$P^* \$ (2) = & -0.1864720673458945*GRAV*AA(2)*M(2) - \\ & 0.1864720673458945*GRAV*AA(2)*M(3) - \\ & 0.1864720673458945*GRAV*M(2)*XBAR(2) + \\ & 0.946406099039542*AA(2)*M(2)*YBAR(2) \end{aligned}$$

$$\$P^* \$ (3) = 1.*GRAV*M(3)*ZBAR(3)$$

$$\$P^* \$ (4) = SFTOR(2)$$

$$\$P^* \$ (5) = SFTOR(1)$$

$$\$P^* \$ (6) = BN2(1)$$

$$\$P^* \$ (7) = SFTOR(3)$$

$$\$P^* \$ (8) = BN2(2)$$

$$\$P^* \$ (9) = BN2(3)$$

$$\begin{aligned} \$P^* \$ (10) = & 0.5510350569644098*AA(2)*DD(3)*M(3) + \\ & 0.5510350569644098*AA(2)*M(3)*YBAR(3) + \\ & 0.6266743428547712*AA(2)*M(2)*ZBAR(2) \end{aligned}$$

$$\begin{aligned} \$P^* \$ (11) = & -0.4431256774282322*AA(2)*DD(3)*M(3) - \\ & 0.4431256774282321*AA(2)*M(3)*YBAR(3) + \\ & 0.7792812509020994*AA(2)*M(2)*ZBAR(2) \end{aligned}$$

$$\begin{aligned} \$P^* \$ (12) = & -0.3804528033808052 - \\ & 0.1383863766233774*KYY2(1)*M(1) - \\ & 0.2122473170855674*AA(2)**2*M(2) - \\ & 0.02967614257656331*KXX2(2)*M(2) - \\ & 0.1087102340468141*KYY2(2)*M(2) - \\ & 0.1035370830387533*KZZ2(2)*M(2) - \\ & 0.2122473170855674*AA(2)**2*M(3) - \\ & 0.2750146732518451*AA(3)**2*M(3) - \\ & 0.1383863766233774*DD(3)**2*M(3) - \\ & 0.1016742198909498*KXX2(3)*M(3) - \\ & 0.2383025165194176*KYY2(3)*M(3) - \\ & 0.03671215673242757*KZZ2(3)*M(3) - \\ & 0.4244946341711351*AA(2)*M(2)*XBAR(2) - \end{aligned}$$

$$\begin{aligned} &0.5500293465036903*AA(3)*M(3)*XBAR(3) - \\ &0.2767727532467548*DD(3)*M(3)*YBAR(3) \end{aligned}$$

$$\$P^* \$ (13) = -1.*GRAV*M(2)*YBAR(2)$$

$$\begin{aligned} \$P^* \$ (14) = &0.03954977001903916 + \\ &0.2287973483287534*KYY2(1)*M(1) + \\ &0.157235038938721*AA(2)**2*M(2) + \\ &0.03533004941978912*KXX2(2)*M(2) + \\ &0.1934672989089642*KYY2(2)*M(2) - \\ &0.03623225997024321*KZZ2(2)*M(2) + \\ &0.157235038938721*AA(2)**2*M(3) - \\ &0.2694525088995099*AA(3)**2*M(3) + \\ &0.2287973483287536*DD(3)**2*M(3) + \\ &0.1888279826418586*KXX2(3)*M(3) - \\ &0.3094218745864047*KYY2(3)*M(3) + \\ &0.03996936568689477*KZZ2(3)*M(3) + \\ &0.3144700778774421*AA(2)*M(2)*XBAR(2) - \\ &0.5389050177990198*AA(3)*M(3)*XBAR(3) + \\ &0.4575946966575072*DD(3)*M(3)*YBAR(3) \end{aligned}$$

$$\begin{aligned} \$P^* \$ (15) = &-0.2456155982889842 - \\ &0.01624226265009617*KYY2(1)*M(1) + \\ &0.0939741253167811*AA(2)**2*M(2) - \\ &0.0889235810932947*KXX2(2)*M(2) + \\ &0.07268131844319859*KYY2(2)*M(2) + \\ &0.02129280687358251*KZZ2(2)*M(2) + \\ &0.0939741253167812*AA(2)**2*M(3) + \\ &0.09165564756849*AA(3)**2*M(3) - \\ &0.01624226265009611*DD(3)**2*M(3) - \\ &0.5650489420980276*KXX2(3)*M(3) - \\ &0.4571510318794415*KYY2(3)*M(3) + \\ &0.5488066794479313*KZZ2(3)*M(3) + \\ &0.1879482506335623*AA(2)*M(2)*XBAR(2) + \\ &0.18331129513698*AA(3)*M(3)*XBAR(3) - \\ &0.03248452530019222*DD(3)*M(3)*YBAR(3) \end{aligned}$$

$$\begin{aligned} \$P^* \$ (16) = &-0.4739123690332002 + \\ &0.2363456990274378*KYY2(1)*M(1) - \\ &0.178009252988034*AA(2)**2*M(2) + \\ &0.115279798595574*KXX2(2)*M(2) + \\ &0.1210659004318637*KYY2(2)*M(2) - \end{aligned}$$

$$\begin{aligned}
& 0.2990751534198976 * KZZ2(2) * M(2) - \\
& 0.178009252988034 * AA(2) ** 2 * M(3) + \\
& 0.1423809061828166 * AA(3) ** 2 * M(3) + \\
& 0.2363456990274378 * DD(3) ** 2 * M(3) + \\
& 0.14200589877006 * KXX2(3) * M(3) + \\
& 0.04804110592543898 * KYY2(3) * N(3) + \\
& 0.0943398002573777 * KZZ2(3) * M(3) - \\
& 0.3560185059760682 * AA(2) * M(2) * XBAR(2) + \\
& 0.2847618123656332 * AA(3) * M(3) * XBAR(3) + \\
& 0.4726913980548757 * DD(3) * M(3) * YBAR(3)
\end{aligned}$$

$$\$P^* \$ (17) = -1. * AA(2) * M(3) * ZBAR(3)$$

$$\begin{aligned}
\$P^* \$ (18) = & -0.5430206388065662 - \\
& 0.0963713135584128 * KYY2(1) * M(1) + \\
& 0.1676138420435843 * AA(2) ** 2 * M(2) - \\
& 0.4491414325181287 * KXX2(2) * M(2) + \\
& 0.3527701189597157 * KYY2(2) * M(2) - \\
& 0.1851562769161311 * KZZ2(2) * M(2) + \\
& 0.1676138420435845 * AA(2) ** 2 * M(3) + \\
& 0.005511963003661525 * AA(3) ** 2 * M(3) - \\
& 0.0963713135584128 * DD(3) ** 2 * M(3) + \\
& 0.1285618779164423 * KXX2(3) * M(3) + \\
& 0.2304451544785172 * KYY2(3) * M(3) - \\
& 0.2249331914748557 * KZZ2(3) * M(3) + \\
& 0.335227684087169 * AA(2) * M(2) * XBAR(2) + \\
& 0.01102392600732305 * AA(3) * M(3) * XBAR(3) - \\
& 0.1927426271168257 * DD(3) * M(3) * YBAR(3)
\end{aligned}$$

$$\begin{aligned}
\$P^* \$ (19) = & 0.7071067811865475 * GRAV * AA(3) * M(3) + \\
& 0.7071067811865474 * GRAV * M(3) * XBAR(3)
\end{aligned}$$

$$\$P^* \$ (20) = 1. * DD(3) * M(3) * ZBAR(3)$$

$$\begin{aligned}
\$P^* \$ (21) = & -0.5233546696798645 + \\
& 0.0114877786792432 * KYY2(1) * M(1) + \\
& 0.1093526052990443 * AA(2) ** 2 * M(2) + \\
& 0.4276053002863393 * KXX2(2) * M(2) - \\
& 0.4161175216070962 * KYY2(2) * M(2) + \\
& 0.5254701269061404 * KZZ2(2) * M(2) + \\
& 0.1093526052990442 * AA(2) ** 2 * M(3) + \\
& 0.001895616432506327 * AA(3) ** 2 * M(3) +
\end{aligned}$$

$$\begin{aligned}
& 0.01148777867924333*DD(3)**2*M(3) + \\
& 0.0913817666053604*KXX2(3)*M(3) + \\
& 0.0817896043586238*KYY2(3)*M(3) - \\
& 0.07989398792611714*KZZ2(3)*M(3) + \\
& 0.2187052105980888*AA(2)*M(2)*XBAR(2) + \\
& 0.003791232865012653*AA(3)*M(3)*XBAR(3) + \\
& 0.02297555735848666*DD(3)*M(3)*YBAR(3)
\end{aligned}$$

$$\$P^* \$ (22) = 1.*AA(3)*M(3)*ZBAR(3)$$

$$\begin{aligned}
\$P^* \$ (23) &= 0.707106781186547*AA(2)*AA(3)*M(3) + \\
& 0.707106781186547*AA(2)*M(3)*XBAR(3)
\end{aligned}$$

$$\begin{aligned}
\$P^* \$ (24) &= 0.5773502691896258*AA(3)*DD(3)*M(3) + \\
& 0.5773502691896258*DD(3)*M(3)*XBAR(3) + \\
& 0.5773502691896258*AA(3)*M(3)*YBAR(3)
\end{aligned}$$

B.6 Physical Values Used

GRAV = 9.80665
 M(1) = 12.96
 M(2) = 22.37
 M(3) = 6.97
 AA(2) = 0.4318
 AA(3) = -0.0191
 DD(3) = 0.1505
 XBAR(2) = -0.3289
 XBAR(3) = 0.01466
 YBAR(2) = 0.0050
 YBAR(3) = 0.00845
 ZBAR(2) = 0.2038
 ZBAR(3) = 0.1101
 KXX1 = 0.1816
 KYY1 = 0.0152
 KZZ1 = 0.1811
 KXX2 = 0.0596
 KYY2 = 0.1930

KZZ2 = 0.1514

KXX3 = 0.0783

KYY3 = 0.0786

KZZ3 = 0.0021

BN2(1) = 4.5

BN2(2) = 3.5

BN2(3) = 3.5

SFTOR(1)= 5.95

SFTOR(2)= 6.82

SFTOR(3)= 3.91